DRAGON USER

The independent Dragon magazine

October 1986

Contents

Editorial

Last month's bargain cards have been stripped as if by locusts, and dealers with cheap hardware report that it's going fast. We have so many questions to answer that we're running a page of Communication this month. New subscriptions are still flowing in, and we get plenty of suggestions as well.

Sometimes we can pull out of the hat the very article which scores of readers have been requesting; sometimes we can't but we keep trying. We would like to do that round up of printers, but we won't publish anything which hasn't been tested by a Dragon User — so if you have recently installed a new printer, get in touch. You may be able to help DU.

Do you write? As well as original ideas from contributors, we often have a pile of suggestions looking for a micro-wright. Tell me who you are and what you can do, and I'll send you the next requests list.

I'll be setting aside a page for programming tips soon, so we want to hear from anyone who thinks their solution is neater and more elegant (or just more useful) than the run of the mill. This can be planning and designing as well as routines. The Dragon Can Do, so tell us what it's doing for you.

Telephone number
(All departments)
01-437 4343

Editor
HELEN ARMSTRONG

Production Editor
BARBORA HÁJEK

Associate Editor
JOHN COOK

Editorial Secretary
ANNE MARIE O'DWYER

Advertisement Manager
SIMON LANGSTON

Administration
GERALDINE SMYTH

Managing Editor
PETER WORLOCK

Publishing Director
JENNY IRELAND

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues
ISSN 0265-177. Telex: 296275
Dragon User, 12/13 Little Newport Street, London WC2H 7PP
Published by Scot Press Ltd.

Typesetting and Production by Artext Limited, London NW1.
Printed by Headley Brothers Ltd., Ashford, Kent.
Registered at the Post Office as a newspaper.
Dragon and its logo are trademarks of Eurohard Ltd.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

Seeing is deleting

I HAVE discovered that the Dragon in EDIT-mode saves the rest of the text in the cursor position whenever you hit the "I"-key and get into INPUT-mode.

This gives you the facility of deleting unwanted characters by using the left arrow key instead of counting n characters for nD. You simply move the cursor to the first character you wish to retain after the characters to be deleted, then enter INPUT-mode and finally hit the left arrow key until the first character to be retained is reached. This gives you visible control of what you're deleting.

Tommy Strand
Hammerveien 130
N-8600 MO I RANA
Norway

Line feed switch

MY interest was aroused by "New Line" in *Dragon Answers*, August 1986. The Seikosha GP100A Mark 2, which I used, has four small switches inside the cover. When the printer leaves the factory, all four switches are in the OFF position. The third switch controls line feed, and is put in the ON position to obtain line feed. This is dealt with in the maker's manual, and mentioned in an article in DU July 1983, page 17.

Eric Troop
25 Anfield Road
Bolton
Lancs BL3 3DA

FC error found

THERE is nothing quite like using information from one's own article and finding it to be in error . . . I was setting up a numeric POKE value in a BASIC program for use in a machine code routine as per the technique on page 21 of the March 1986 DU, and four values larger than 32767 an FC error occurs when using the 'AND 255' technique at the top of the page. For unsigned double byte values that may exceed 32767, the least significant byte of, say, variable N can be obtained by:

POKE P2+1, N—(INT (N/256)*256)

Signed values—1 to—32768 will suffer a similar fate so

POKE P2+1, TEMP—(INT(TEMP/256)*256

will be needed. This was the technique I had always used until adopting the shorter 'AND 255' statement!

Pam D'Arcy
21 Wycombe Lane
Woodburn Green
High Wycombe

Ideas for Israel?

I AM the head manager of the Israeli Dragon Users Club. Our club is a youth club; we wanted to join an adult club, but there is no such club in Israel, so we decided to build our own club. We would like to hear from our Dragon users with ideas on how to run a club.

Itamar Sagey
Hersel 11/7
Kfar-Saba 44455
Israel

Helpful sounds

IN THE course of writing some simple Basic programs for children with mental handicaps, a recurring problem was the inability to read instructions on the screen by children who were well able to follow verbal instructions. A solution was found in recording the instructions verbally on the tape directly after the program, timing the speech with a stop-watch. Then all that is needed is a MOTOR ON: AUDIO ON command in the program, followed by a suitable pause, provided by the Dragon's inbuilt timer, or a fournext loop. After some experimentation the lengths of the loops were adjusted as follows:

1 to 1500 = 2 seconds; 1 tp 3550 = 5 seconds; 1 to 7050 = 10 seconds; 1 to 10600 = 15 seconds; 1 to 14150 = 20 seconds; 1 to 20500 = 30 seconds.

A for-next loop must be used if the timer has already been set as part of the program. If the pause is set slightly longer than the instructions then the MOTOR OFF:AUDIO OFF command can be followed by a tone to indicate when to start the game (just like those horrible telephone answering machines). Of course, reminders to switch off the 'play' button on the recorder would be a good idea as well.

Using this method, perhaps some 'real' sounds could be incorporated into games (without the pause). I rigged a cassette with a continuous loop to play some recorded galloping noises for a horse-race game, the possibilities for adding sounds, music or speech into programs are only limited by the length of the cassette tape using this method, unless a continuous loop is rigged in a separate cassette.

Stuart Beardwood
1 West Bottom
Hob Lane
Norland
Sowerby Bridge
West Yorks

Elusive computer

IN THE editorial from September 1985 you mentioned that Eurohard were trying to produce a 128K Dragon with hard disks and Flex. Do you have any more information? Also, way back in July 1984 GEC brought out a new Dragon, called the Professional, and a couple of people were advertising it for £699 inc. VAT. When I rang Cotswold Computers last year they told me the Professional never made it to the open market. Is this true or is there any chance of obtaining one?

C.C. Bailey
9 Draycote House
Birch Hall Lane
Longsight
Manchester

EUROHARD have not produced any new models that we know of in the last twelve months; the only effective way of getting a 128K Dragon is with a hardware add-on. We had pictures of the Professional many months ago, but we never saw a live one. If any did make it on the market, they will be in private hands by now. Does anyone know more about the Professional?

Lack of data

IN THE August edition you featured Screen Designer by Dennis Riley. An interesting and rewarding program. There was one snag, however. Those of us who do not possess an Alldream assembler and had to POKE in the data found that the 'Shifting Screen' featured in Listing 4 would not work because of insufficient data.

On close close examination I find that the following appears to have been omitted:

06BB 73 63 72 65 65 6E
06BB 73 63 72 65 65 6E
06CC 00
06D2 62 61 72 80 74 6F

In the main assembler listing (2), two items appear to have been 'lost'.

7B6E 32
7BD4 72

I wish you and Dragon User every success and I hope that you obtain the support you deserve, not only from subscribers, but from advertisers as well.

Ray Smith
5 Glen Road
Parkstone
Poole
Dorset

Change of address

THANK YOU for publishing my article for the Touchmaster Graphics Tablet. As I have moved, the address in the article is no longer correct.

For those who still write to ask, I can supply any of my programs from Dragon user at the original price of £3.00 each. These include the Wordprocessor (disc/tape), machine code monitor, Graftext and the graphics tablet program.

Peter Whittaker
c/o 126 Coventry Road
Nuneaton
Warwickshire
CV10 7AD

Cult status

MANY thanks for your mention in the August edition of DU of the efforts I and my colleagues put in at the Osset show with the amateur radio demo.

Comments made in DU about the Dragon being almost a 'cult' m/c amongst hams are certainly true. I have come to rely on my three Dragons quite considerably in my amateur radio activities as have others I know of. Some time ago DU carried a letter from (or an article about) the very large group of hams in the Nottingham/Derby area, who regularly every Sunday operate on a frequency of 144.525 MHz FM with news, graphics and general data being sent all over the place, well, they are still going strong and I hope, subscribing to the mail order DU.

Keep up the good work. Incidentally, we have already had several enquiries from hams, one in South Africa, as a result of the August issue. Did you know that *Rommels Revenge* from Design Design was written by a ham? If you think this is all coincidence, have a look at the American Co-Co mag, *Rainbow*. Over the last three years virtually every edition carries articles and programs concerning amateur radio applications. Knowing something about electronics, you see, we all know that the 6809 is superior to the Z80, and all the rest. That is why we bought 'em, not to play games.

Finally, talking of Rainbow, an ad. appeared in there some years ago for an add-on board (with Z80) to enable the Co-Co to run CP/M80. Interesting, eh?

P.J. Read G6ZZE
7 Fairview Avenue
Whetstone
Leicester LE8 3JQ

Sulking designer

I THANK you for publishing my program "SCREEN DESIGNER", in the August edition; however, an error has been pointed out to me that will cause the program to 'Hang Up', or 'sulk' during Option 5, if a program title of 8 characters is used.

This is because of the program, instead of being sent to the ROM subroutine 8H8006, it is sent to the routine CBLINK, and therefore because of the following instruction, 'A' cannot meet the condition to allow the program to continue.

To rectify this then, in the Machine Code Data listing &H0BEF should be '80 and &H0BF0 should be '06'.
The Assembler listing should read:
7B0C BD8006 GTNMW2 JSR $8006.

I apologise for any confusion and inconvience this error may have caused.

May I also through this page extend my apologies to anyone who had to wait for a copy of the program, or a reply, due to my spending a lot of time away during the August-September period.

Dennis Riley
21 Colmore Road
Wortley
Leeds LS12 4DF

Routine solution

In response to Brian Cadge's request (DU, August '86), I have worked out a few CUMANA 2.0 DOS POKEs and the addresses of the sector READ/WRITE routines in ROM, and here they are:

OOEB	number of active DRIVE.
OOEE:OOEF	buffer address (for sector read/write).
OOF6	if nonzero-decrement 0605 in each IRQ.
0605	when reaches zero — turn off the disk motors.
0609	VERIFY flag: 0=off, nonzero=on.
060A	DRIVE number
0697:0698	AUTO current line number.
0699:069A	AUTO increment.
069B	AUTO flag: 0=off, nonzero=on.
069C:069D	ERROR GOTO line number.
069E	ERROR GOTO flag: 0=off, nonzero=on.
069F:06A0	ERL.
06A1	ERR.
E56D	sector read routine.
E643	sector write routine.

ENTRY for both routines: IX HI=track, IX LO= SECTOR−1, OOEB and OOEE set to the relevant values before calling these routines.

After using these routines, remember to POKE a nonzero value to location 0605, because if you don't, the motor won't stop (it is stopped only when this location REACHES zero, not when it already is zero).

Danny Halamish
Ophra 90906
Israel.

Random numbers

OVER the past couple of months a lot has been said about random numbers, mostly in the competition. Very little information is provided in the manual about using RND; however, I recently came across a 16 bit random number generator in a machine language package. My first question was, why not use a ROM call to the RND function, my second was how does the Basic RND function work?

The first answer was easy. The Basic RND function returns a random number in the Floating Point Accumulator as a real number (hence in theory it can have any value 1E-37 to 1E+38). This is naturally not easy to handle in machine code except by using the INTCNV routine in Basic ($8B2D).

The second answer was more interesting. From what I gleaned from the ROM the RND function seems to be based on a 32-bit random number generator linked with Floating Point mathematics. For some time I have used "RND (−TIMER)" to scramble the RND function for games programs etc. without knowing why it worked. Looking at the RND function it becomes obvious that this has three separate functions.

In the manual RND is said to return a number between 1 and the number specified. In actual fact RND(X) gives a whole random number between 1 and X if X is greater than 0, a fractional random number between 0 and 1 if X=0 and if X is greater than 0 the value of the new seed (this will also be between 0 and 1). It appears that the seed is generated from a function of the number 1088834987.

This means that a set sequence of random numbers can be repeatedly reproduced by using the same seed, without having to store them in an array and could be useful for statistical studies:

```
10 A=RND(−1)20 FORX⊕=OTO9:PRINTRND(10):NEXTX.
```

This program will always produce the same sequence as the same seed is generated each time.

Since a program will never be run at exactly the same time after the computer has been turned on every go, the value of TIMER will be effectively random. So seeding the RND function with the value of TIMER will guarantee a different sequence each time.

A common suggestion for scrambling the RND function is based on the 'wait for a key' basis. It becomes obvious that this merely shifts the sequence so many places along.

To truly randomise RND the only solution is to use RND(−TIMER).

Robin J. Telkman
1 Sycamore Street
Sale
Cheshire

Bridging a gap

I have a Dragon 32 and a Bridge master. I inserted the cartridge before switching on the computer, and wiped the contents of the cartridge off. The shop which supplied me does not exist any more. The receiver of Dragon Data Ltd. could not help me.

Dr. David J. Unwin
51 Maryville Park
Belfast
BT9 6LP

User Group

THE National Dragon Users Group tell us that their monthly newsletter Dragon Update is up to its 24th issue, and that their membership is increasing steadily since they formed in 1984 to combat rumours that the Dragon was dead.

The NDUG has members overseas as well as in the UK, and encourages other newsletters to reprint their articles and reveiws. Dragon Update prints reviews, news, practical information about software and hardware, and readers' programs. "Anyone in the group is likely to be picked on at any time to do just about anything . . . we try to balance things so that we cover all Dragon related interests, so that there is something for everyone . . . the subscription covers only essentials such as printing, postage, phone calls etc."

Subscriptions are £8 annually, £9 overseas, from Paul Grade (Chairman),6 Navarino Road, Worthing, Sussex. NDUG is usually represented at national 6809 shows, so look for them next time.

Disc editing utility

PAMCOMMS have produced a new utility for editing machine code programs. *Sourcemaker* disassembles memory to DragonDOS disc in the form of a DSKDREAM Source Code Data File, which can then be loaded into DSKDREAM for subsequent editing and reassembly. It is ideal, says Pam D'Arcy, for tailoring the likes of the hex dumps which appear in DU to one's own system and personal tastes. It is also helpful in editing software, particularly in changing from cassette to a disc based system, where support is not available from the *Sourcemaker* with User Guide and a Master and Work discs is available at £9.00 inland, £10.00 overseas, all inclusive, and as a cassette based system shortly, at the same prices. This version will work in conjunction with all the DREAM packages (ALLDREAM cartridge, cassette, DeltaDOS, DREAM cassette, DREAM and DREAM BUG cassette).

Orders and enquiries to Pamcomms Ltd., 21 Wycombe Lane, Wooburn Green, High Wycombe, Bucks HP10 0HD.

Radio Dragon

PEAKSOFT tell Dragon User that they have assured supplies of Dragon power supplies and individual transformers. Says Harry Whitehouse: "There has been a shortage of the original product for some time. As a result, we commissioned the design and manufacture of a new, heavy duty replacement. As a service to the Dragon community, which has supported us for the best part of four years, we sell them at quite a heavy loss, in order that everyone can keep their Dragons in action." The complete power supply costs £16.95, with two years' guarantee. For information contact Peaksoft at 48 Queen St., Balderton, Newark Notts NG24 3NS. Tel: (0636) 705230.

Peaksoft also have a service of interest to radio amateurs: *Radio Dragon* is an electronic magazine which is on the air for 30 hours a week. Dragon and Tandy users with modems can download the whole magazine with a 2-3 minute cheap rates phone call, and then save, print or just read it through. It's free of charge, and does not take paid advertising, but will mention new services.

Not Washed

PART of a line is missing from line 890 of last month's Core Ware: the missing characters are <ot crashed> to complete the phrase 'The rest have not crashed.'

Dragon dance

MARIDIAN, the software house which produced Mazerace and Spellbox last year, have three new packages on the market.

"Underbeings of Croth" is an adventure which begins in a deep pit where a hapless adventurer is imprisoned in an iron cage. You must rescue him and guide him to the surface. The game uses upper and lower case text, 86 locations and 165 words, and can be saved. The price is £3.95.

"Decathlon" is a ten-game competition for one to five players, including Anagram, Hangman, Shootout, Spider, etc. The games are designed to suit people of different abilities and are good for parties, claim the makers. The price is again £3.95.

Something a little different for the music buff: 'accurately transcribed' four part harmony renditions of music by Bach and Scott Joplin, which can be played through the Dragon using its own sound capabilities, and routed through a hifi if the user wishes. The two Bach and one Joplin tapes are £3.95 each, £8.95 the set.

Orders and enquiries to Maridian, Birchmore Cottage, Nairdwood Lane, Prestwood, Great Missenden, Bucks HP16 OQQ.

Show releases

BLABY COMPUTER Games have added a fruit machine game to the three new releases (*Trun*, *Boulder Crash* and *Temple of Doom*) which they are releasing at the Dragon Computer Show in Cardiff in October (information about the show from John Penn computing on Bordon (04203) 5970). "And anything else we have finished at the time" says John Bailiss.

Blaby also have a cache of working and non-working Dragons to sell, which they will be bringing to the shows in London (November) and Cardiff; they do transformers as well.

Knechtel Knews

BERND KNECHTEL Software are introducing a new price (and method of payment) for their Disk Support Tool (DST) package on disk. The new price is £7.00 inc. postage and packing; payment can now be made by postal order (Bernd is not entirely clear whether this includes cheques and money orders; if sending a Post Office postal order, it should be crossed for security) instead of direct transfer.

We hope to be printing a review of DST soon. When ordering, please state which version of DragonDOS you are using, and attach your name and address separately to both your order and your postal order.

Order and enquiries to Bernd Knechtel, Arnold-Willhelm-Str. 9, 5630 Remscheid, West Germany.

Even cheaper

PROOPS DISTRIBUTORS, who last month put a collection of working and non-working Dragon 32s up for sale at their north London warehouse, report that the units have been selling well, and that they are slashing the price of the non-working, spares-only units still left to £10 each.

The warehouse is at the Haybridge Estate, Castle Road, Camden Town, London NW1; for information and phone (credit card) sales phone (01) 267 6911.

Dragon date

OWING to the change in distribution, Dragon User will now be delivered in the last week of the month prior to the cover date.

News Extra — Page19

Dragon User People's Chart

DALLAS, Texas: in the multi-storey offices of Ewing Oil, wide shouldered Personal Secretaries glide silently between wide, gleaming executive desks across wide expanses of whispering carpet. In the panelled suite occupied by J. R. Ewing, J.R. himself bites down hard upon a wide cigar as he peruses a file deposited by one of these creatures.

"Whaddya mean, takeover bid? Who in hell is this Wintersoft, anyway? Part of ICI? Goddam nerve!"

"Sir," breathes the vision. "The latest report has just arrived. Finance are concerned that Wintersoft may be in a stronger position than we imagined."

"Like what?"

"Well, sir, *Juxtaposition - Barons of Ceti V* has topped the Dragon User People's Chart for the third month ..."

At this point J.R. bites his cigar in half. J.R.'s nerves will be jangled still further by Mr. Robert Glassman of Dorset, who boldly states "Save fuel, get a Dragon, it's energy for life!". Mr. Glassman wins our anagram contest.

You, too can win £25-worth of software from Microdeal if you can devise an intimidatingly witty sentence from the letters (or some of them) in the titles of your favourite three Dragon software packages. To do this, of course, you have to list your favourites - five, in fact, on that form to your right (or copy it) and send them to us. It's a gas!

Results September 1986

1 Juxtaposition.......................(Wintersoft)
2 Shock Trooper.....................(Microdeal)
3 Total Eclise.................(Eclipse Fenmar)
4 Moon Cresta........................(Incentive)
5 Syzygy ....(Microdeal)

Chart Eight

Voting for Chart No. 8 closes at 1pm on Friday, 17th October 1986. Entries recieved after that time will not be eligible for inclusion in that month's voting. The editor's decision is final. Only one entry per individual per month will be allowed.

My top 5: Voting Month 7

1 ....
2 ....
3 ....
4 ....
5 ....

Name ....
Address ....
....
....
....
....

My phrase is:..

Dragonsoft

New software for review should be sent to *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Fifty and out...

Program: *Cassette 50*, **Supplier:** Cascade Games, 1-3 Haywra Crescent, Harrogate, N. Yorks.
Price: £9.95

WHEN this cassette came out many eons ago, I remember thinking to myself, "it can't be? 50 games on one cassette? They must all be atrocious!" Through time though, I mellowed and gained a different view of humanity. I felt that maybe humanity was inherently good, maybe the cassette has some good stuff on it.

A few months later saw me buying the 'Fabulous' Cassette 50.

I rushed home, set up Digby (my computer) and loaded the first game, *Maze eater*. It wasn't very good. In fact it was written in BASIC and didn't really look very nice. in fact it was awful. But what did I care... I still had 49 games to go.

I loaded the second game, *Galactic Attack*. It wasn't really very good, in fact it was written in BASIC and didn't really look very nice. In fact it was awful. But what did I care... I still had 48 games to go!

Three hours later I loaded the last game, *Exchange*. It wasn't really very good, etc.

I was not a happy person. Digby and myself had had a great time loading fifty totally useless games.

All the games were in BASIC, although some were not in Dragon BASIC — one of the games would not run as it was written in ZX-81 BASIC.

No attempt whatsoever was made at text formatting so the games looked as bad as they played!

May I congratulate Cascade Software on producing not one or two but fifty unplayable games.

Jason Orbaum

Justified writer

Program: *Electronic Author*
Supplier: Smithson Computing **Price:** £14.95 cassette, £19.95 disk

THIS review is like no other I have ever written, for it was written on a new and better word-processor than the dated *Telewriter*. Yes, sure, *Telewriter* is a fine word Processor, it's just that Smithson's *Electronic Author* is better.

The format is text window and command window. The text window is either 51 or 64 characters wide dependant on user choice, and twenty lines deep. The command window is two lines and allows input of the abbreviated commands to control I/O functions and text manipulation.

The keyboard scan doesn't appear to miss letters, ever, and the text, even with 64 characters a line is clear, neat, and pleasant to read.

In theory, the B-side's "Config" program should allow the WP to work with any printer, it works fine with my Seikosha, but I'm having trouble trying to get the centring to work!! It's a tape based WP but a DragonDos version is available and a Delta Version is currently in production.

All the usual features: Search and replace (selective and global), block move, copy, and delete, word count, etc. are very easy to operate. The program also has one up on an un-patched *Telewriter* in that it has a right justification mode built into it so that it is possible to have your text printed in clear blocks without a jagged right hand margin.

All in all, I like this word-processor and will now be using it instead of *Telewriter*. It would be nice to see word-wrap incorporated as an option.

Jason Orbaum

Communication

Send in your questions, requests, and pleas to Communication, Dragon User, 12-13 Little Newport Street, London WC2.

Problem: Can anyone help us with software/hardware/ special adaptations, suitable for use by children with physical and mental handicaps? We have been given a Dragon 32 and cassette recorder, but cannot find any of the above. We call our Dragon "Puff", can someone provide some magic? Can he be sound-operated by wiring up a microphone? Can he talk to us? Any offers of help/ideas gratefully accepted.
Enquirer: Stuart Beardwood, c/o Westgarth Children's Home, 79 Victoria Road, Elland, West Yorkshire.

Problem: What are the functions of the ROM routines at addresses 36925, 40253, 34029, 34932? What are the Tandy equivalents?
Enquirer: Paul Marlow, 50 Lime Avenue, Bentley, Walsall, West Midlands WS2 0JP.

Problem: Could anybody help me in securing a circuit diagram for an EPROM curner for the Dragon 32?
Enquirer: Phillip D. Kenney, 49 Craven, Stonydelph, Tamworth, Staffs B77 4HH.

Problem: Is there anyone in Edinburgh who has a Dragon 32K who is willing to exchange games, etc?
Enquirer: Ryan Henderson, 2 East Clapperfield, Edinburgh, Scotland EH16 6TU.

Problem: Is there anyone out there who is willing to print out listings of Basic programs for a Dragon User with no printer?
Enquirer: T. Jenkins, Llys Helys, Nowport, Pembrokeshire SA42 0QZ.

Problem: Does anyone have, or know about, the Superchoc cartridge? Does anyone know of a TRS/Dragon converter for MIC. Pross., does it exist or is it a myth?
Enquirer: Ian Rockett, 2 Knowle Road, Burley, Leeds, Yorkshire LS4 2PJ.

Problem: Is anyone using a Dragon 32 for duplicate bridge scoring?
Enquirer: I.W. Cain, 18 Tilbury Road, Rainham, Kent ME8 7PX.

Problem: I need to buy or borrow (all post paid) the following Dragon Users: June 1983, July 1983, August 1983, September 1983; also, has anybody got the demonstration 1-7 from Sprite Magic, I keep getting mistakes in lines 20 or 30 and I need one that works.
Enquirer: Dennis Gates, 194 Bek Road, Newton Hall Est., Durham, DH1 5LH.

Problem Could you please tell me how to achieve a scrolling effect in Basic, on the Hi-res screen, as in games such as Skramble, Speed Racer etc.
Enquirer Steve Driver, 93 Valley View, Lemington, Newcastle-upon-Tyne.

Problem Would coupling two Dragon 32s together produce extra memory, ie more than the 25k already available? If so, how do I do the link up?
Enquirer J. D. Wall, Stone Gables, Clarbeston Road, Dyfed SA63 4SL.

Problem I have owned a Smith Corona Speedtext 80 printer since Christmas, and have found it to be a bargain at around £100, however the manual is not clear on the production of graphics. I have tried several screen dumps and none of them work satisfactorily. I would also like to know how to load a program from DragonDOS using the Boot command.
Enquirer Shawn Barker, 24 Carroll Drive, Longton, Stoke-on-Trent, Staffs ST3 1SF.

Problem I have got Speed Racer, but because I have a self centering joystick, so when I try to get track 2, I just get track 1, 3 or 4 instead. If anyone has any hints please write.
Enquirer D. Minty, c/o WO2 (SSM) Minty, H. Q. Squadron, 7th Sig. Regt., BFPO 15.

Problem I have a CPA-80 printer, but I don't have a program for dumping graphics onto it the right way up. Please help me because it is not very useful having a printer with pictures sideways on it.
Enquirer Nicholas Parkes, 15 Battlefield Road., St. Albans, Herts AL1 4DA.

Problem Is there any way I can adapt Microdeal's Backgammon game to work with my Trojan Light Pen, as the game has a light pen facility?
Enquirer Daniel Snell, 41 College Cresc., Oaley, Aylesbury, Bucks HP18 9QZ.

Problem I have a Tandy CGP-115. How do I dump the hi-res screen to the printer, is it a matter of a few simple pokes, or do I have to buy some software? If so, where do I buy it?
Enquirer D. S. Philips, 10 Broadstone hall Road South, South Reddish, Stockport, Cheshire SK5 7DQ.

Problem I have a Dragon 32 and an alphacom printer with an interface for a Spectrum. Can anyone tell me if the printer is compatible with the Dragon, and how I can wire it to a suitable socket? I have a centronics cable, but when I remove the interface from the printer I will be left with a bundle of wires and I don't know how to mate them up with the cable. Can anybody send me a wiring diagram or help. I will pay expenses.
Enquirer R. Burrows, 134 Parkfield St., Rusholme M/c14 7PT.

Problem I have a small program of about 2k which is in Basic and occupies lines 1 to 67 consecutively and which has a large number (400+) of data lines
attached to it, starting at line 100, making the total memory in excess of 31k. I used POKE 25,6:NEW to enable this large use of data. I tried to 'RENUM 100,100,1' the whole program, and got RO ERROR. My four year old manual did not help, and I could not retrieve the program and had to CLOAD again. I checked the memory and found 60 left. By removing some data lines and leaving 500 bytes in momory, the RENUM worked. Could anyone explain what RO ERROR is and how to avoid it, using all available memory?
Enquirer Alastair Burr, 156 Folly Lane, St. Albans, Herts AL3 5JG

Problem Could anybody help a TRS CoCo 32k owner with the conversation of the Writer program (May '85) and the Window program (July '85) from Dragon format, especially the EXEC ADDRESS or DATA statement changes. I would also like to know if anyone is willing to sell their July 1986 issue. ((*You and 10,000 others — Ed.*))
Enquirer Edward Neave, 31 Albeton Lane, Severn Beach, Bristol BS12 3PP.

Problem Is there a Stockport or South Manchester Dragon Club?
Enquirer Keith Emmerson, 31 Waverton Av., Heaon Chapel, Stockport, Cheshire SK4 5JT.

Problem I am looking for a copy of the Computavoice voice synthesiser program by Dragon Data.
Enquirer John Campbell Rees, 12 Stuart Street, Trehervert, Rhondda, Mid Glamorgan, CF4 25PR.

Problem I have a Dragon 32 converted to 64, and I am puzzled by not being able to find the maximum (?) memory in 64 Mode; also I have been given a Hitachi compact floppy disc drive model HFD 3055 (Byte Drive) and was wondering if anyone knew how to use it with the Dragon.
Enquirer Mr. W. E. Patmore, 35 Lodge Oak Lane, Tonbridge, Kent TN9 2EQ.

Problem I am a frustrated Dragonite with no printer. There used to be adverts for people who would print out listings for you, but I can't find any now. Do you know anyone with a suitable printer who will do this service?
Enquirer T. Jenkins, Llys Helyg, Newport, Pembs. SA42 0QZ.

Problem What is RTTY and could I have some information on amateur radio?
Enquirer R. Vaughan, 189 Port Tennant Road, Port Tennant, Swansea, W. Glamorgan SA1 8JU.

Communication

Stuck for a routine? Need some obscure equipment? Feeling cut off? Fear not — someone, somewhere can help you! Write down your problem on the coupon below (make it as brief and legible as possible) together with your name and address and send it to Communication, Dragon User, 12/13 Little Newport Street, London WC2H 7PP. We'll publish it as soon as we can — meanwhile, maybe there's someone you can help this month!

Problem ....
....
Name ....
Address ....
....

Disc rescue

***Pam D'Arcy** rescues good sectors from a corrupted disc.*

HAVING purchased a secondhand pair of 40 track single-sided drives at a bargain price (despite a round trip of 250 miles), a spate of RF errors followed making me feel that perhaps they were not such a good bargain after all. This was a particularly badly timed problem as it prevented essential BACKUPs being taken of discs full of hex dumps of some contract work that I was testing on the Dragon. Of necessity, I quickly produced a few lines of BASIC to copy all the readable sectors of an affected disc to a newly formatted disc on the second drive. I have since added additional options, including single drive working, to produce DISKBUGS (DISK Back Up Good Sectors). Although it will now form part of my DISK—KIT package (£10 + 50p P&P — hence the 'credit' lines in it), it is such a short but useful program that I cannot resist sharing it even with non-purchasers of DISK-KIT.

It has only been tested on 40 track single-sided, single and dual drives as I cannot get our 80 track double-sided system to function satisfactorily on the Dragon, but optional parameters are built into it for 40/80 single/double sided single/multiple drive operation. In order to cater for single drive users, 72 consecutive sectors (4 single-sided drive tracks) of data are read into X$ and Y$ arrays between disc swaps, thus requiring 10 disc changes for a 40 track single-sided back-up — not much worse than BACKUP itself and pretty good, I feel, for a BASIC program. The occasional apparent 'hesitation' in reading in BASIC's garbage of the string stack taking place (72 sectors needs 18K of string stack at a time).

When reporting disc errors, no screen paging is included for those without printers. The error reports scroll up from the bottom of the screen, there being ample time for them to be noted on paper before they disappear should there be a large number of them.

To ensure that the destination disc is error free before commencing the back up, DSKINIT is a pre-copy option. Users of single sided 80 track systems will recognise lines 360-400 as being the guts of the 'patch program' supplied by Dragon Data to correct the wrongly set up sector bit map.

A sector of ASCII space characters is written to the destination disc in place of sectors unable to be read from the source disc. Should the faulty sector have occurred in the middle of a BASIC program, LOADing and LISTing the program from the new disc after DISKBUGS has finished will probably show a wildly excessive line number preceding a number of space filled lines on the screen. DELete the offending line number and as much as possible of an affected BASIC program will have been rescued.

When initially typing/testing programs containing automatic error handling (eg. line 20 ERROR GOTO800), it is often easier to start off by temporarily REMarking out such lines, although it is obviously an essential part of the program once you are happy that it seems to be typed in correctly. DISKBUGS program listing attached.

```
10 REM DISKBUGS/09JUN86 (C)PAMCOMMS LTD, 21 WYCOMBE
LANE, WOOBURN GREEN, HIGH WYCOMBE, BUCKS.HP10 0HD
(BOURNE END 26232) dragondos back up good sectors
20 ERROR GOTO800
30 PMODE0,1:PCLEAR1:CLEAR22000
40 DIM SD,DD,TT,ST,DR,S,T,D,SR,P,F,K,K$,A$,B$,E$(2)
50 T=0:SR=18*4:DIM X$(SR),Y$(SR)
60 E$(0)="    ":E$(1)="RD.":E$(2)="WR."
70 F=0:GOSUB720
80 PRINT"PRINTER READY (Y/ANY) ?        ";
90 GOSUB750
100 IF K$="Y" THEN P=2 ELSE P=0
110 PRINTK$
120 PRINT:PRINT"SOURCE      DRIVE NUM IF<>1:";:GOSUB780
130 IF K$<>"2" AND K$<>"3" AND K$<>"4" THEN K$="1"
140 K=VAL(K$):PRINT K
150 SD=K
160 PRINT"DESTINATION DRIVE NUM IF<>2:";:GOSUB780
170 IF K$<>"1" AND K$<>"3" AND K$<>"4" THEN K$="2"
180 K=VAL(K$):PRINT K
190 DD=K
200 PRINT"80 TRACK DRIVE (Y/ANY) ?    ";
210 GOSUB750
220 IF K$="Y" THEN TT=80 ELSE TT=40
230 PRINTTT;
240 PRINT"DOUBLE-SIDED DRIVE (Y/ANY) ? ";
250 GOSUB750
260 IF K$="Y" THEN ST=36 ELSE ST=18
270 PRINTK$
280 PRINT:PRINT"DSKINIT DEST. DISK (Y/ANY) ? ";
290 GOSUB750
300 PRINTK$:IF K$="N" THEN 420
310 GOSUB720
320 PRINT@104,"initialise disk"
330 GOSUB680
340 PRINT:PRINT:PRINT"DISK BEING INITIALISED NOW":
DSKINIT DD,(ST/18),TT
350 IF TT=80 AND ST=18 THEN360 ELSE410
360 SREAD DD,20,1,A$,B$
370 A$=MID$(A$,1,90)+STRING$(38,CHR$(255))
380 B$=STRING$(52,CHR$(255))+STRING$(72,CHR$(0))
+CHR$(80)+CHR$(18)+CHR$(255-80)+CHR$(255-18)
390 SWRITE DD,16,1,A$,B$
400 SWRITE DD,20,1,A$,B$
410 DIR DD
420 IF SD=DD THEN DR=1 ELSE DR=2
430 GOSUB650
440 CLS
450 SS=1
460 S=1
470 F=1:PRINT@0,"READING DR."SD;"TR.";T;"SE.";S
```

```
480 SREAD SD,T,S,X$(SS),Y$(SS)
490 SS=SS+1:IF SS=SR+1 THEN 510
500 S=S+1:IF S=ST+1 THEN T=T+1:GOTO460 ELSE GOTO470
510 F=2:IF DR=2 THEN540
520 PRINT@0:PRINT:PRINT@0,"";:GOSUB680
530 PRINT@32
540 SS=1
550 T=T+1-(SR/ST)
560 S=1
570 F=2:PRINT@0,"WRITING DR.";DD;"TR.";T;"SE.";S
580 SWRITE DD,T,S,X$(SS),Y$(SS)
590 SS=SS+1:IF SS=SR+1 THEN 610
600 S=S+1:IF S=ST+1 THEN T=T+1:GOTO560 ELSE GOTO570
610 T=T+1:IF T=TT THEN PRINT"    diskbugs backup
completed":GOTO850
620 IF DR=2 THEN450
630 PRINT@0:PRINT:PRINT@0,"";:GOSUB660:PRINT@32
640 GOTO450
650 PRINT:PRINT"       back-up good sectors"
660 PRINT"PUT SOURCE       DISK IN DRIVE";SD;
670 IF DR=1 THEN690
680 PRINT"PUT DESTINATION DISK IN DRIVE";DD;
690 PRINT"AND PRESS ANY TO COMMENCE";
700 GOSUB780
710 RETURN
720 CLS:PRINT"  diskbugs by pamcomms limited"
730 PRINTSTRING$(32,"=")
740 RETURN
750 GOSUB780
760 IF K$<>"Y" THEN K$="N"
770 RETURN
780 K$=INKEY$:IF K$=""THEN780
790 RETURN
800 IF F=1 THEN D=SD ELSE D=DD
810 PRINT@(15*32),E$(F);"ERR.";ERR;"DR.";D;"TR.
";T;"SE.";S
820 IF P=2 THEN PRINT#-2,E$(F);" ERROR";ERR;"DR.
";D;"TR.";T;"SE.";S
830 IF F=1 THEN X$(SS)=STRING$(128," "):Y$(SS)
=STRING$(128," "):GOTO490
840 IF F=2 THEN 590
850 CLEAR200:END
```

Fault Line

Create a landscape with this program by ***P. Whittaker***

BACK IN the February issue of Dragon User, I wrote an article on Three-Dimensional Graphics for the Dragon. Although I dealt with the equations needed for the rotation and manipulation of 3D objects, I did not cover the subject of hidden line removal. As a result, the objects drawn tended to look as though they were made of chicken-wire, and were often hard to understand. To draw more solid looking objects we need to devise a way of removing the lines which show through from the back of the object, and mess up the display. Such a technique is called 'Hidden-Line-Removal'.

Flat surface

FAULT LINE MACHINE is a program which generates artificial landscapes from a flat surface, and demonstrates one easy method of 'Hidden Line Removal'. The program works by generating a number of random fault lines across the face of the landscape, and moving the opposite sides of the fault-line either up or down a small distance. After sufficient faults have been developed, a hilly landscape will develop. If the process is continued, the landscape will continue to get more and more pointed and mountainous.

When RUN an options list will be displayed, press the (G) key to generate a new landscape. The computer will then ask for a grid size for the landscape. This can be up to about 25 on a Dragon 32, but can be higher if on a Dragon in 64K mode. Then the computer will ask for the number of faults to generate. The landscape is drawn on the PMODE4,5 graphics screen, and displayed on the PMODE4,1 screen. The display is updated each time the results of a new fault-line are calculated. Whilst the landscape is being generated the display will be in green, but once the final landscape has been drawn, the display will change to white. However, at this stage, the display will still be in the usual 'wire frame' style, and will be see-through. The hidden-line removal can be selected later. Pressing the (M) key will return to the menu screen, whilst any of the other keys will call the appropriate routines.

If the landscape has not developed sufficiently, press the (C) key, and you can generate a few more fault lines.

Once a landscape has been generated, it can be saved to disc by pressing the (S) key. It can be reloaded at a later date by pressing the (L) key. When the landscape data is saved, the first piece of data is the size of the arrays. This is so that the load routine can cope with different sized landscapes. When the data is reloaded, the program will know what size of array it is dealing with.

The remaining functions all deal with the details of the landscape display. Pressing the (F) key will fill in the sea, at all locations below the baseline level of the landscape. The landscape will be redrawn using hidden-line removal, but this time the tide will have come in, and the deepest valleys will have disappeared. This will often make it easier to get a feel for the shape of the land. In a similar way, pressing the (R) key will also redraw the landscape using hidden-line removal, but this time will draw it with the tide out.

Fig.1 Black lines to paint out a segment

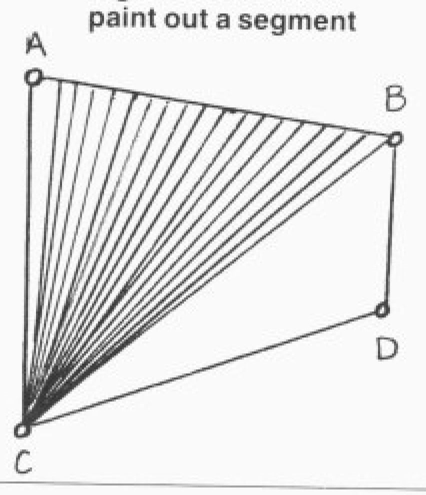
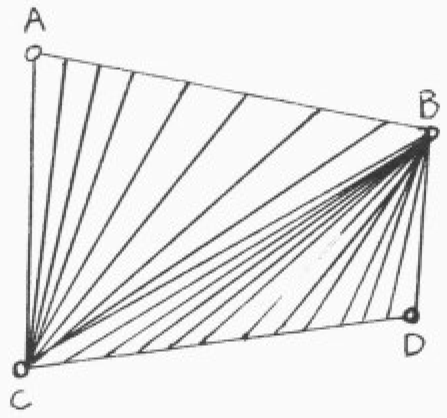


Fig.2 Segment fully painted Any background obliterated



If the landscape has developed in such a way that it is all either under the sea, or all above it, then it can be raised or lowered by pressing the (V) key, and then entering the vertical offset for the display. When this recalculation is finished, the programme will return to the menu screen. To draw in the new landscape, you will need to press either the (F) or (R) keys.

Once you are happy with the landscape, the display can be saved by pressing the (D) key. This is currently set up for disc, so tape users will need to change the command from SAVE A$,3072,9216, 41194 to CSAVEM A$,1536,7680,41194.

The program makes use of two sets of arrays (x(),Y(), and X2(),Y2()). These are dimensioned at the start of the program. If you are using a Dragon without discs, or in 64K mode, these arrays can be made bigger, which will allow for a larger grid to be used. The X and Y arrays are used to store the landscape data, whilst the X2 and Y2 arrays are used to store the modified landscape data used when drawing the display.

When a landscape is generated (160-) the program first asks for the grid size, and the number of faults. A flat surface is calculated to fill the grid (220-270), and then this surface is manipulated by each of the faultlines. This generated grid is calculated to hold the screen co-ordinates of the landscape, and these co-ordinates are directly manipulated by the program. This will save us having to make later perspective calculations. Each faultline is generated by randomly choosing two points on the surface. From these points (X1, Y1) and (X2,Y2) we can calculate the equation of the faultline, AX+B=Y (340-360). Then another couple of random numbers are picked to decide which side of the lines will be moved up or down. (380-430) The computer will then scan through the arrays and modify the vertical component of each point it finds on the appropriate side of the fault line.

The equation for a straight line uses the formula AX+B=Y. With two values for X and Y, we can calculate the values of A and B, and so derive the equation of the line itself. A=(Y1-Y2)/(X1-X2) and B=Y1-(A*X1). Then, once we have this line equation, we can test all of the points on the grid surface to see if they fall above or below the line. Inserting the X value in the equation, will generate a Y value. If this is greater than the Y value of our co-ordinate, then we know that it it below the line. Similarly, if it is smaller, then the co-ordinate is above the line. In this manner, we can discover on which side of the fault line each co-ordinate is, and move it up or down accordingly.

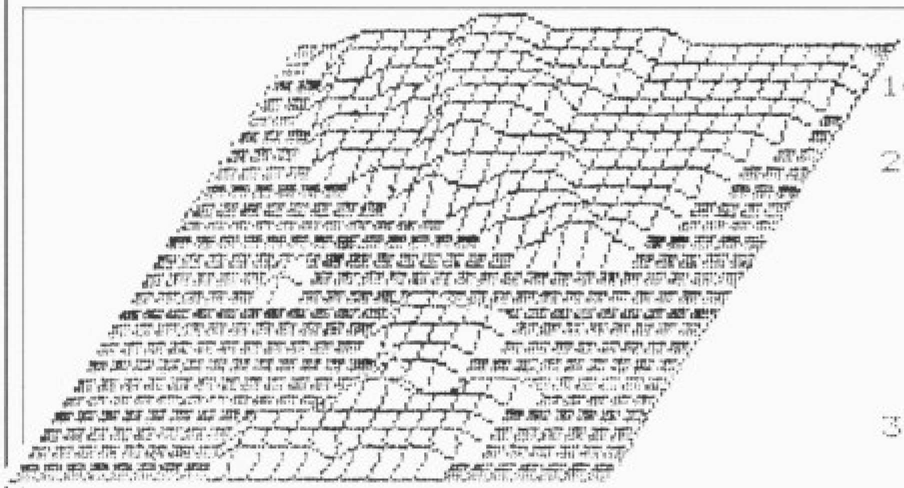
The landscape is then drawn out on the PMODE4,5 screen, and then displayed to the PMODE4,1 screen (450-550). When the landscape is drawn in as a wire frame, the programme uses the main arrays X() and Y(). When the landscape is drawn using hidden line removal the secondary arrays X2() are used. This is because the (F)ill sea routine modifies the array data, and would not allow for the (R)emove or (C)ontinue functions if it used the main arrays.

Routines

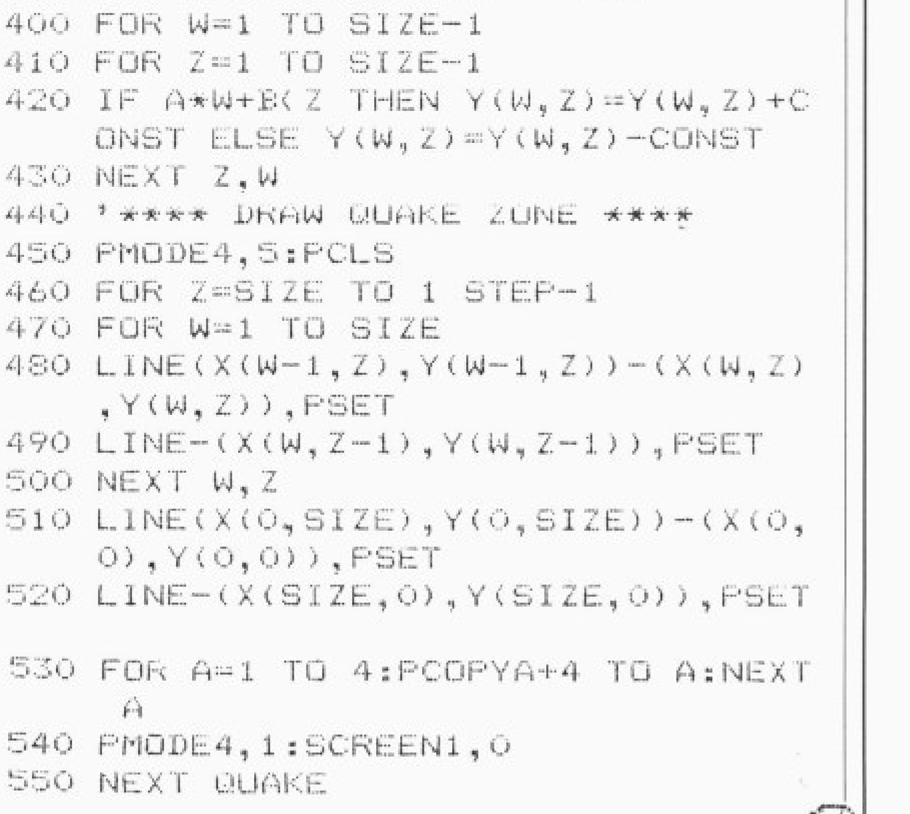
The (F)ill routine reads through the data in the main arrays (850-920), and if it finds any co-ordinates which fall below the grid base line, it sets them to that base line level. All of these co-ordinates are copied into the secondary arrays for display. The (R)emove routine works by copying the original arrays across into the secondary display arrays. Both of these routine lead automatically into the 3D-solid display routine (580-810). This draws in the landscape starting with the furthest back parts, and working

towards the front of the display. In this way, the closer parts of the landscape are drawn over the areas behind, and any hidden lines get removed. To make sure that these hidden lines are indeed covered, the landscape is not drawn in as a wire frame, but as a solid. The space between each line of the wire frame is filled with black lines, painting out anything underneath.

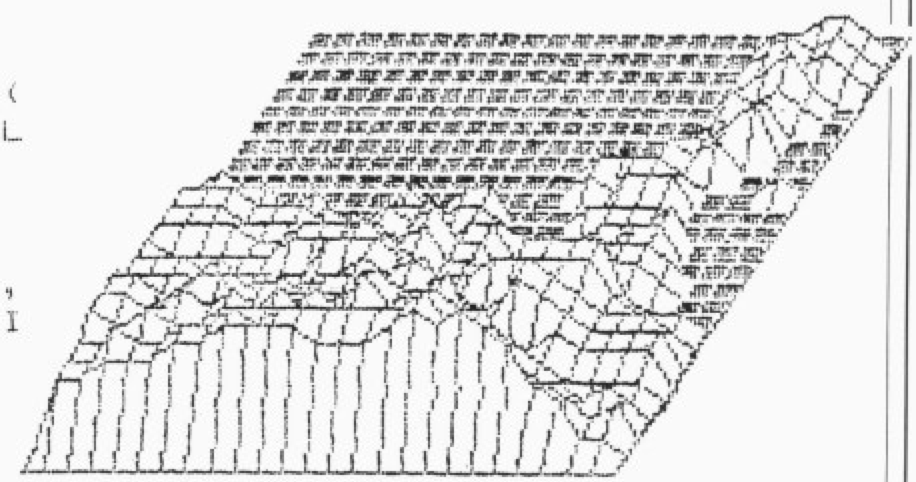
This is done by once again using the Y=AX+B equation. This time we use the corner co-ordinates for the two points from which we generate the line equation (Fig. 1). The program will first calculate the equation of the line from corner 'A' to corner 'B'. Then it will draw a black line from each point on that line to corner 'C'. (This is done by inserting the range of possible X values between corners 'A' and 'B' into the line equation.) When this is done, the program will work out the equation for the line between 'C' and 'D' and then draw a black line from each point on this line to corner 'B' (Fig. 2). Then finally, the program will draw in the outline 'A-B-D-C-A' in white. By doing this, and working from the back of the display object towards the front, we can make any object appear solid. We are in effect displaying a solid black object with white contour lines on its surface to make it visible. Using this technique, it should be possible to draw any object on screen, giving it a solid appearance.



```
10 PCLEAR8:DIM X(25,25),Y(25,25),X2
   (25,25),Y2(25,25)
20 CLS:PRINT@6,"FAULT LINE MACHINE"
   :PRINT:PRINT"G....GENERATE FAUL
   T LINES.":PRINT"C....CONTINUE F
   AULT GENERATION.":PRINT"S....SA
   VE LAND MASS DATA.":PRINT"L....
   LOAD LAND MASS DATA.":PRINT"F..
   ..FILL IN SEAS."
30 PRINT"R....REMOVE SEAS.":PRINT"D
   ....SAVE DISPLAY.":PRINT"V....V
   ERTICAL OFFSET."
40 A$=INKEY$:IF A$="" THEN 40
50 IF A$="M" THEN 20
60 IF A$="G" THEN 160
70 IF A$="C" THEN PMODE4,5:INPUT"NU
   MBER OF EXTRA FAULTS";Q:GOTO280
80 IF A$="S" THEN 970
90 IF A$="L" THEN 1060
100 IF A$="D" THEN 1160
110 IF A$="F" THEN 860
120 IF A$="R" THEN 940
130 IF A$="V" THEN 830
150 GOTO20
160 POKE65495,0
170 A=RND(-TIMER)
180 CLS:PRINT@265,"FAULT LINE MACHI
   NE":PRINT
190 PMODE4,5:PCLS
200 INPUT"SIZE OF GRID";SIZE
210 INPUT"NUMBER OF FAULT LINES";Q
220 FOR A=0 TO SIZE:FOR B=0 TO SIZE
230 V=120/(SIZE)
240 H=2*230/(SIZE*3)
250 Y(A,B)=150-B*V
260 X(A,B)=10+A*H+B*H/2
270 NEXT B,A
280 SCREEN1,0
290 FOR QUAKES =1 TO Q
300 X1=RND(SIZE-1):X2=RND(SIZE-1):Y
   1=RND(SIZE-1):Y2=RND(SIZE-1)
310 IF X1=X2 THEN 300
320 IF Y1=Y2 THEN 300
330 IF X1=X2 THEN 300
340 '****EQUATION = AX+B=Y ****
350 A=(Y1-Y2)/(X1-X2)
360 B=Y1-(A*X1)
370 '**** UP OR DOWN ****
380 C=RND(2):IF C=2 THEN CONST=-3 E
   LSE CONST=+3
390 '*+ CALCULATE EARTHQUAKE **
400 FOR W=1 TO SIZE-1
410 FOR Z=1 TO SIZE-1
420 IF A*W+B<Z THEN Y(W,Z)=Y(W,Z)+C
   ONST ELSE Y(W,Z)=Y(W,Z)-CONST
430 NEXT Z,W
440 '**** DRAW QUAKE ZONE ****
450 PMODE4,5:PCLS
460 FOR Z=SIZE TO 1 STEP-1
470 FOR W=1 TO SIZE
480 LINE(X(W-1,Z),Y(W-1,Z))-(X(W,Z)
   ,Y(W,Z)),PSET
490 LINE-(X(W,Z-1),Y(W,Z-1)),PSET
500 NEXT W,Z
510 LINE(X(0,SIZE),Y(0,SIZE))-(X(0,
   0),Y(0,0)),PSET
520 LINE-(X(SIZE,0),Y(SIZE,0)),PSET
530 FOR A=1 TO 4:PCOPYA+4 TO A:NEXT
   A
540 PMODE4,1:SCREEN1,0
550 NEXT QUAKE
```



```
560 PMODE4,1:SCREEN1,1:GOTO40
580 'DRAW IN AS 3D-SOLID
590 FOR Z=SIZE TO 1 STEP -1
600 FOR W=1 TO SIZE
610 X1=X2(W-1,Z):X2=X2(W,Z):X3=X2(W,Z-1):X4=X2(W-1,Z-1)
620 Y1=Y2(W-1,Z):Y2=Y2(W,Z):Y3=Y2(W,Z-1):Y4=Y2(W-1,Z-1)
630 A1=(Y1-Y2)/(X1-X2):B1=Y1-(A1*X1)
640 IF K=1 THEN IF Y1=Y2 AND Y1=150-Z*V THEN COLOR0,1 ELSE COLOR1,0
650 FOR X=X1 TO X2
660 Y=A1*X+B1
670 LINE(X,Y)-(X3,Y3),PRESET
680 NEXT X
690 A1=(Y4-Y3)/(X4-X3):B1=Y4-(A1*X4)
700 FOR X=X4 TO X3
710 Y=A1*X+B1
720 LINE(X1,Y1)-(X,Y),PRESET
730 NEXT X
740 LINE(X1,Y1)-(X2,Y2),PSET:LINE-(X3,Y3),PSET:LINE-(X4,Y4),PSET:LINE-(X1,Y1),PSET
750 NEXT W,Z
760 COLOR1,0
770 LINE(X(0,0),Y(0,0))-(X(SIZE,0),Y(SIZE,0)),PSET:LINE-(X(SIZE,SIZE),Y(SIZE,SIZE)),PSET
780 POKE65494,0
790 SOUND100,1
800 SCREEN1,1
810 GOTO40
820 'MOVE LANDMASS
830 PRINT:INPUT"ENTER LAND SHIFT OFFSET";A
840 FOR W=1 TO SIZE-1:FOR Z=1 TO SIZE-1:Y(W,Z)=Y(W,Z)-A:NEXT Z,W:GOTO20
850 'FILL IN THE SEA
860 SCREEN1,0:FOR W=0 TO SIZE
870 FOR Z=0 TO SIZE
880 IF Y(W,Z)>150-Z*V THEN Y2(W,Z)=150-Z*V ELSE Y2(W,Z)=Y(W,Z)
890 X2(W,Z)=X(W,Z)
900 NEXT Z,W
910 K=1
920 PMODE4,1:PCLS:SCREEN1,0:GOTO590:'DRAW LAND MASS
930 'REMOVE SEA
940 SCREEN1,0:FOR W=0 TO SIZE:FOR Z=0 TO SIZE
950 Y2(W,Z)=Y(W,Z):X2(W,Z)=X(W,Z):NEXT Z,W:K=0
960 PMODE4,1:PCLS:SCREEN1,0:GOTO590
970 CLS:PRINT@230,"SAVE LANDSCAPE DATA":PRINT:PRINT
980 INPUT "FILE TITLE";FILE$
990 FWRITE FILE$;SIZE
1000 FOR A=0 TO SIZE:FOR B=0 TO SIZE
1010 FWRITE FILE$;X(A,B)
1020 FWRITE FILE$;Y(A,B)
1030 NEXT B,A
1040 CLOSE
1050 GOTO20
1060 CLS:PRINT@230,"LOAD LANDSCAPE DATA":PRINT:PRINT
```



```
1070 INPUT"ENTER FILE TITLE";FILE$
1080 FREAD FILE$;SIZE
1090 FOR A=0 TO SIZE:FOR B=0 TO SIZE
1100 FREAD FILE$;X(A,B)
1110 FREAD FILE$;Y(A,B)
1120 NEXT B,A
1130 CLOSE
1140 V=120/SIZE
1150 GOTO20
1160 PRINT:INPUT"SAVE DISPLAY TITLE";A$
1170 SAVE A$,3072,9216,41194:GOTO20
1180 '(C)1986 BY PETER WHITTAKER.
```

Mission impossible

***David Berry** sets his sights on a high-flying foe.*

THE FIRST listing is nothing more than a set of instructions; there's no reason on Earth for anyone to type it in: just read and digest. The second is the Basic part of the program which does all the slow speed setting up and so on. Be careful to type the first two lines exactly as I have them or you'll get trouble saving high scores which are stored in the second line in place of all those zeroes. Each time you get a new high score resave the Basic part of the program (My highest was 2480). The final block is the machine code.

Rapier is a shooting gallery. You're sitting in the rear gun turret of some aircraft and in front of you is a gunsight which can be controlled from the keyboard or through the joystick. Across your field of view, just in range of your missiles, flies an endless stream of enemy aircraft. Press the fire button and after a delay your missile is launched and starts tracking towards the centre of your sight. Remember that: the missile has no interest in life other than finding the centre of the sight. When it gets there it explodes *whether or not the sight is fixed on an aircraft at the time.* One tip I'll give you is that if you wait until you've fixed the sight on an aircraft before firing you'll almost always miss. You have to fire the missile then fix the aircraft and follow it through until the missile arrives which requires a nice bit of co-ordination.

Listing 1 — Instructions

```
100 REM INSTRUCTIONS FOR RAPIER.
110 REM CLOAD"RAPIER"  then  RUN
120 REM The machine code will be
130 REM loaded by the programme.
140 REM Use a joystick in the RH
150 REM port.  The  stick  moves
160 REM the sight and  the  fire
170 REM button launches the miss
180 REM -ile. The missile always
190 REM tracks towards the centr
200 REM of the sight. The  speed
210 REM at which the sight moves
220 REM is variable depending on
230 REM the position of the sti-
240 REM ck.
250 REM Alternatively  the  key-
260 REM board can be  used  with
270 REM the O P @ and space keys
280 REM for left, up, right  and
290 REM down; the 1,2 and 3 keys
300 REM to  set  the  speed  and
310 REM ENTER to fire.
320 REM 'X' will pause the  game
330 REM and 'H'will halt it.
```

Now and again the enemy will launch a counter-attack and fire a missile at you. These start off as pretty docile, easy to hit objects, but speed up as your score improves until, frankly, you don't have a chance of hitting them. Then they hit you and you lose a life.

At the start you have three lives and can add another four by knocking out everything in sight before your score reaches 100. After that you're on the downward path. High flying aircraft are easier to hit since you have more time to fix your sight and so they score less than low flying ones. If your sweat starts clogging up the joystick hold 'X' down to pause the game; if you get suicidal press 'H' to end it all.

The Basic routine gives you a nice analysis at the end so you can see how badly you've done; whether your hit/miss ratio is improving and so on. To have another go hit ENTER. Stopping is by the RESET button, I'm afraid.

If you don't want to type it all in (who can blame you?) send me a tape, return postage and two quid (per tape) to bribe the wife with and you'll get a copy straight out of the computer. Please, please, have the tape positioned where you want Rapier put and allow plenty of room as we'll not go hunting for odd corners of space on your tapes. You also need to say which side of the tape to put it on and whether you want a duplicate set on the same tape.

David Berry, 6 Vanbrugh Close, High Cross, ROGERSTONE, Gwent. NP1 0DF.

Listing 2 — Basic

```
10 GOTO 30: (C) S.BERRY 1983
20 (SYSPARAMS::000000:000000:000000)
30 IF PEEK(&H7001)<>&H71 THEN GOSUB 670
40 CLEAR 200,&H6EFF
50 GOSUB 840
60 PCLEAR 8
70 PMODE 3
80 '
90 SCREEN 1,0
100 COLOR 2,3
110 PCLS 3
120 GOSUB 300
130 GOSUB 200
140 PCOPY 1 TO 5
150 PCOPY 2 TO 6
160 PCOPY 3 TO 7
170 PCOPY 4 TO 8
180 GOSUB 360:CLS
190 GOTO 80
200 REM DRAW TURRET
210 LINE(59,28)-(196,164),PSET,B
220 LINE(0,0)-(59,28),PSET
230 LINE(255,0)-(196,28),PSET
240 LINE(59,164)-(0,191),PSET
250 LINE(196,164)-(255,191),PSET
260 LINE(0,0)-(255,191),PSET,B
270 LINE(119,180)-(136,188),PSET,B
280 PAINT(120,181),1,2
290 RETURN
300 REM STARS
310 FOR I=0 TO 99
320 C=RND(4):IF C=1 THEN C=3
330 PSET((RND(255)),(RND(191)),C)
340 NEXT
350 RETURN
360 DATA &H7000,&HB6,&H7201,&H80,&H7202,&H60,&H7203,00,&H7680,00,&H7681,00,&H771
0,00,&H7711,00,&H7832,00,&H79A2,03,&H6F70,0,&H6F71,0,&H6F72,0,&H7847,03,&H7848,&
HE8,&H7853,01,&H7854,&HF4,&H785F,00,&H7860,&HFA,&H786B,00,&H786C,&H7D,&H7877,06,
&H7878,&H3E
370 DATA &H7709,&H08,&H770A,&HFF,&H71B0,&H40,&H79A1,00
380 FOR K=1 TO 27
```

```
390 READ I,J:POKE I,J:NEXT
400 POKE &H7100,RND(255)
410 POKE &H7101,RND(255)
420 EXEC &H76D0:EXEC &H76FC
430 EXEC &H7060:EXEC &H7000
440 POKE &HFFD6,0
450 PAINT(0,0),1,1
460 AD=&H3600+PEEK(&H7E00)
470 CLS6:PLAY"T100;ABA":PRINT @0,"            ANALYSIS"
480 MISS=PEEK(&H7680)*256+PEEK(&H7681)
490 HIT=PEEK(&H7710)*256+PEEK(&H7711)
500 MHY=PEEK(&H79A1)
510 SCORE=VAL(HEX$(PEEK(&H6F70)))*10000+VAL(HEX$(PEEK(&H6F71)))*100+VAL(HEX$(PEE
K(&H6F72)))
520 IF MISS=0 THEN RATIO=0 ELSE RATIO=HIT/MISS
530 PRINT @64,"     TOTAL HITS RECORDED:";HITS
540 PRINT @128,"    TOTAL MISSES RECORDED:";MISS
550 PRINT @192,"       HIT/MISS RATIO:";:PRINT USING "£££.££";RATIO
560 HS=0
570 FOR I=AD TO AD+5
580 HS=HS*10+(PEEK(I)-&H30)
590 NEXT I
600 IF SCORE>HS THEN HS=SCORE:GOSUB 740
610 PRINT @448,"        HIGH SCORE:";HS
620 PRINT @256,"TOTAL ENEMY MISSILES FIRED:";MHY
630 PRINT @384,"  YOUR FINAL SCORE WAS:";SCORE
640 PRINT @320,"    ENEMY MISSILES YOU HIT:";MHY-PEEK(&H7832)
650 IF INKEY$<>CHR$(13) GOTO 650
660 PLAY"T100;ABA":RESTORE:RETURN
670 CLS0:S$=CHR$(128):F$=STRING$(32,CHR$(236)):PLAY"T100;ABA"
680 PRINT @32,"load";S$;"machine";S$;"code";S$;S$;S$;S$;S$;"start";S$;"tape";F$
690 PRINT @96,"use";S$;"joystick";S$;"in";S$;"r";CHR$(124);"h";S$;"port";S$;"or"
;S$;S$;"use";"o";S$;"p";S$;CHR$(96);S$;"and";S$;"space";S$;"keys";S$;"for";S$;"l
eft";S$;"up";"right";S$;"and";S$;"down";STRING$(18,S$);F$
700 PRINT @224,"horizontal";S$;"sight";S$;S$;"speed";S$;S$;"depends";"on";S$;"th
e";S$;"position";S$;"of";S$;"the";S$;S$;"joystick";"or";S$;"is";S$;"controlled";
S$;"by";S$;"the";S$;S$;"one";S$;"two";"and";S$;"three";S$;"keys";STRING$(18,S$);
F$
710 PRINT @384,"press";S$;"x";S$;"for";S$;"pause";S$;"and";S$;"h";S$;"for";S$;"h
alt";"at";S$;"any";S$;"time";STRING$(21,S$);
720 CLOADM
730 RETURN
740 H$=STR$(HS):H$=RIGHT$(H$,LEN(H$)-1):J=6
750 H$="0"+H$
760 IF LEN(H$)<6 GOTO 750
770 FOR I=AD+5 TO AD STEP-1
780 I$=MID$(H$,J,1)
790 K=ASC(I$)
800 POKE I,K
810 J=J-1
820 NEXT
830 RETURN
840 CLS6:PLAY"T100;ABA"
850 PRINT @00,"              MENU"
860 PRINT @64,"  J = JOYSTICK CONTROL"
870 PRINT @128,"  K = KEYBOARD CONTROL"
880 PRINT @256,"  E = EASY LEVEL"
890 PRINT @320,"  D = DIFFICULT LEVEL"
900 PRINT @384,"  I = IMPOSSIBLE LEVEL"
910 PRINT @192,"      SELECT CONTROL ?"
920 I$=INKEY$:IF I$<>"J" AND I$<>"K" GOTO 920 ELSE PRINT @215,I$:PLAY"T100;ABA"
930 IF I$="J" THEN POKE &H7205,&H8E:POKE &H7206,&H01:POKE &H7207,&H5B ELSE POKE
&H7205,&H7E:POKE &H7206,&H7B:POKE &H7207,&H30
940 PRINT @448,"      SELECT LEVEL ?"
950 I$=INKEY$:IF I$<>"E" AND I$<>"D" AND I$<>"I" GOTO 950 ELSE PRINT @469,I$:PLA
Y"T100;ABA"
960 IF I$="E" THEN POKE &H7650,16:POKE &H7665,8:POKE &H7E00,&H2D
970 IF I$="D" THEN POKE &H7650,12:POKE &H7665,6:POKE &H7E00,&H34
980 IF I$="I" THEN POKE &H7650,08:POKE &H7665,4:POKE &H7E00,&H3B
990 RETURN
```

```
$ 7000    DATA    ======    *39 71 25 26 05 BD 71 64 20 03 BD 73 04 BD 72 A3
$ 7010    DATA    ======    *BD 70 31 12 12 12 12 B6 72 03 27 03 BD 73 A0 12
$ 7020    DATA    ======    *12 12 BD 7A 34 7E 70 00 00 00 00 00 00 00 00 00
$ 7030    DATA    ======    *04 7A 70 30 26 0F 86 04 B7 70 30 BD 73 50 BD 70
$ 7040    DATA    ======    *8F BD 72 05 39 8E 00 40 30 1F 26 FC 39 00 00 00
$ 7050    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7060    DATA    ======    *CE 1E 00 8E 06 00 10 8E 18 00 EC C1 ED 81 31 3E
$ 7070    DATA    ======    *26 F8 BD 7A C4 BD 79 A4 39 00 00 00 00 00 00 00
$ 7080    DATA    ======    *15 B0 02 55 AA 95 6A A5 5A A9 56 6A 9A A6 A9 BE
$ 7090    DATA    ======    *72 A0 BF 70 80 B6 72 A2 B7 70 82 7A 70 80 7A 70
$ 70A0    DATA    ======    *80 F6 70 82 10 8E 70 8B A6 A5 C6 08 BE 70 80 A7
$ 70B0    DATA    ======    *84 30 88 20 12 12 12 5A 26 F5 7C 70 80 7C 70 80
$ 70C0    DATA    ======    *7C 70 80 C6 08 BE 70 80 A7 84 30 88 20 12 12 12
$ 70D0    DATA    ======    *5A 26 F5 7A 70 80 10 8E 70 83 F6 70 82 58 A6 A5
$ 70E0    DATA    ======    *BE 70 80 30 1D A7 84 30 05 A7 84 31 21 A6 A5 30
$ 70F0    DATA    ======    *1C A7 84 30-05 A7 84 39 00 00 00 00 00 00 00 00
$ 7100    DATA    ======    *6F 3B 8E 71 00 A6 84 49 A8 84 46 6C 01 AB 01 28
$ 7110    DATA    ======    *02 6C 01 A7 84 39 00 00 00 00 00 BD 79 80 20 06
$ 7120    DATA    ======    *FF 57 10 FF 03 FF BD 71 02 B6 71 00 44 8B 20 2A
$ 7130    DATA    ======    *02 80 20 B7 71 21 7F 71 20 44 24 03 7A 71 20 B6
$ 7140    DATA    ======    *71 20 B7 71 25 BE 72 01 FC 71 20 FD 72 01 BD 72
$ 7150    DATA    ======    *A3 BF 72 01 BE 72 A0 BF 71 22 B6 72 A2 B7 71 24
$ 7160    DATA    ======    *39 20 03 12 7A 71 62 26 3F 86 05 B7 71 62 BE 71
$ 7170    DATA    ======    *22 33 89 18 00 EC C4 ED 84 33 C8 E0 30 88 E0 EC
$ 7180    DATA    ======    *C4 ED 84 7C 71 20 27 21 BD 71 45 10 8E 70 83 F6
$ 7190    DATA    ======    *71 24 58 EC A5 BE 71 22 ED 84 30 88 E0 F6 71 24
$ 71A0    DATA    ======    *10 8E 70 8B A6 A5 A7 84 39 7E 71 1B 12 00 00 00
$ 71B0    DATA    ======    *0A 7A 71 B0 27 01 39 86 40 B7 71 B0 CE 78 47 BD
$ 71C0    DATA    ======    *71 D1 33 4C 11 83 78 83 26 F5 CE 77 09 BD 71 D1
$ 71D0    DATA    ======    *39 12 12 12 12 EC C4 54 1C FE 44 24 02 CA 80 5D
$ 71E0    DATA    ======    *26 06 4D 26 03 CC 00 02 ED C4 39 00 00 00 00 00
$ 71F0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7200    DATA    ======    *12 85 7D 00 12 7E 7B 30 10 8E 72 02 8D 0B 30 1F
$ 7210    DATA    ======    *31 3F 8D 05 BD 6F 40 20 4A 34 36 AD 9F A0 0A 35
$ 7220    DATA    ======    *36 A6 84 C6 04 81 3D 2F 04 6C A4 20 2E 81 33 2F
$ 7230    DATA    ======    *06 6C A4 5C 12 20 24 81 29 2F 07 6C A4 5C 5C 5C
$ 7240    DATA    ======    *20 19 81 15 2F 02 20 13 81 0B 2F 07 5C 5C 5C 6A
$ 7250    DATA    ======    *A4 20 08 81 01 2F 02 5C 12 6A A4 F7 70 37 39 12
$ 7260    DATA    ======    *12 12 12 B6 72 01 81 3A 26 03 7C 72 01 81 C5 26
$ 7270    DATA    ======    *03 7A 72 01 B6 72 02 81 1D 26 03 7C 72 02 81 A3
$ 7280    DATA    ======    *26 03 7A 72 02 39 00 00 00 00 00 00 00 00 00 00
$ 7290    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 72A0    DATA    ======    *15 B0 02 86 20 F6 72 02 3D C3 06 00 FD 72 A0 B6
$ 72B0    DATA    ======    *72 01 44 27 0E 80 04 7C 72 A1 26 03 7C 72 A0 81
$ 72C0    DATA    ======    *04 2C F2 B7 72 A2 39 00 00 00 00 00 00 00 00 00
$ 72D0    DATA    ======    *95 6A A5 5A A9 56 AA 55 6A 9A A6 A9 12 12 12 12
$ 72E0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 72F0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7300    DATA    ======    *00 20 03 12 7A 73 02 26 41 86 05 B7 73 02 BE 71
$ 7310    DATA    ======    *22 30 00 33 89 18 00 EC C4 ED 84 33 C8 E0 30 88
$ 7320    DATA    ======    *E0 EC C4 ED 84 7A 71 20 27 21 BD 71 45 10 8E 72
$ 7330    DATA    ======    *D0 F6 71 24 58 EC A5 BE 71 22 ED 84 30 88 E1 F6
$ 7340    DATA    ======    *71 24 10 8E 72 D8 A6 A5 A7 84 39 7E 71 1B 12 00
$ 7350    DATA    ======    *BE 70 80 C6 08 30 89 01 E0 33 89 18 00 A6 C4 A7
$ 7360    DATA    ======    *84 30 88 E0 5A 26 F2 C6 08 30 89 FE 00 33 89 18
$ 7370    DATA    ======    *00 A6 C4 A7 84 30 88 E0 5A 26 F2 BE 70 80 30 1D
$ 7380    DATA    ======    *33 89 18 00 EC C4 ED 84 30 05 33 45 EC C4 ED 84
$ 7390    DATA    ======    *39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 73A0    DATA    ======    *7E 77 70 66 99 99 66 66 99 99 66 66 99 99 66 66
$ 73B0    DATA    ======    *99 12 FF 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 73C0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 73D0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 73E0    DATA    ======    *03 B4 01 12 B6 73 E0 27 07 B7 73 E2 7A 73 E0 39
$ 73F0    DATA    ======    *B6 73 E1 B1 72 02 23 0B 7A 73 E1 B0 72 02 7F 73
$ 7400    DATA    ======    *E2 20 0F 27 0C 7C 73 E1 40 BB 72 02 7F 73 E2 20
$ 7410    DATA    ======    *01 4F 40 8B FF 44 44 44 44 44 44 8B 00 B7 73 E0
$ 7420    DATA    ======    *39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7430    DATA    ======    *01 80 01 12 B6 74 30 27 07 B7 74 32 7A 74 30 39
$ 7440    DATA    ======    *B6 74 31 B1 72 01 23 0B 7A 74 31 B0 72 01 7F 74
$ 7450    DATA    ======    *32 20 0F 27 0C 7C 74 31 40 BB 72 01 7F 74 32 20
```

```
$ 7460   DATA   =======   *01 4F 40 8B FF 44 44 44 44 44 44 8B 00 B7 74 30
$ 7470   DATA   =======   *39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7480   DATA   =======   *B6 74 32 F6 73 E2 3D 27 08 8E 00 01 30 1F 26 FC
$ 7490   DATA   =======   *39 BE 72 01 B6 74 31 B7 72 01 B6 73 E1 B7 72 02
$ 74A0   DATA   =======   *BD 72 A3 BF 72 01 BE 72 A0 BF 74 B6 B6 72 A2 B7
$ 74B0   DATA   =======   *74 B8 7E 74 CD 12 16 10 02 16 10 3F 00 00 00 00
$ 74C0   DATA   =======   *12 12 12 12 BD 74 34 BD 73 E4 7E 74 80 BD 74 E0
$ 74D0   DATA   =======   *BD 75 14 7E 76 04 BD 76 D0 12 12 12 12 39 00 00
$ 74E0   DATA   =======   *BE 74 B9 30 88 E0 33 89 18 00 EC C4 ED 84 30 88
$ 74F0   DATA   =======   *20 33 89 18 00 EC C4 ED 84 30 88 20 33 89 18 00
$ 7500   DATA   =======   *EC C4 ED 84 BE 74 B6 BF 74 B9 39 00 00 00 00 00
$ 7510   DATA   =======   *40 12 12 12 F6 75 10 27 2D 7A 75 10 8E 74 B6 10
$ 7520   DATA   =======   *8E 75 60 F6 74 B8 58 EC A5 30 88 E0 ED 84 30 88
$ 7530   DATA   =======   *40 ED 84 31 A8 08 F6 74 B8 58 EC A5 30 88 E0 ED
$ 7540   DATA   =======   *84 39 12 12 12 12 BE 74 B6 10 8E 75 60 F6 74 B8
$ 7550   DATA   =======   *58 EC A5 ED 84 39 00 00 00 00 00 00 00 00 00 00
$ 7560   DATA   =======   *9A AA A6 AA A9 AA AA 6A 56 AA 95 AA A5 6A A9 5A
$ 7570   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7580   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7590   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 75A0   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 75B0   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 75C0   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 75D0   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 75E0   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 75F0   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7600   DATA   =======   *12 12 12 12 B6 74 31 B1 72 01 27 01 39 B6 73 E1
$ 7610   DATA   =======   *B1 72 02 27 01 39 BD 76 40 B6 76 35 F6 76 36 3D
$ 7620   DATA   =======   *26 09 BD 76 92 BD 70 60 7E 74 D6 BD 77 40 7E 74
$ 7630   DATA   =======   *D6 12 12 12 12 01 01 00 00 00 00 00 00 00 00 00
$ 7640   DATA   =======   *7F 76 35 7F 76 36 B6 74 31 B0 71 20 2A 01 40 81
$ 7650   DATA   =======   *10 23 01 39 12 12 86 01 B7 76 35 B6 73 E1 B0 71
$ 7660   DATA   =======   *21 2A 01 40 81 08 23 01 39 12 12 86 01 B7 76 36
$ 7670   DATA   =======   *39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7680   DATA   =======   *00 0E AB AA 9A BA B0 36 6F 8A A3 4A A2 BE 7F AA
$ 7690   DATA   =======   *12 12 7C 76 81 26 03 7C 76 80 10 8E 76 82 BE 72
$ 76A0   DATA   =======   *A0 F6 72 A2 C1 02 24 02 30 1F 86 07 30 88 A0 EE
$ 76B0   DATA   =======   *A4 EF 84 30 88 20 31 22 4A 26 F4 BD 7A 21 8E 0F
$ 76C0   DATA   =======   *FF 30 1F 26 FC 7E 71 B1 00 00 00 00 00 00 00 00
$ 76D0   DATA   =======   *7F 72 03 8E 77 70 BF 73 A1 C6 39 F7 77 B1 CC 00
$ 76E0   DATA   =======   *00 BD 77 9D 86 7E 12 B7 77 B1 CC B4 01 FD 73 E1
$ 76F0   DATA   =======   *CC 80 01 FD 74 31 86 40 B7 75 10 39 8E 71 64 BF
$ 7700   DATA   =======   *70 06 8E 73 04 BF 70 0B 8E 00 08 BF 78 30 39 00
$ 7710   DATA   =======   *00 88 AA AA AA AA AB EA AD 7A AB EA AA AA AA AA
$ 7720   DATA   =======   *AA AA AA A9 A9 EA AB DA A7 FA A8 3A AA B4 6E A9
$ 7730   DATA   =======   *AB 80 AB C2 AD BE AA A6 B6 BA F0 DA 12 12 12 12
$ 7740   DATA   =======   *12 12 12 12 12 12 12 12 10 8E 77 12 BD 76 9E 10
$ 7750   DATA   =======   *8E 77 20 BD 76 9E 10 8E 77 2E BD 76 9E BD 70 60
$ 7760   DATA   =======   *BD 6F 74 8E 00 20 BF 71 20 BD 76 FC BD 71 1B 39
$ 7770   DATA   =======   *8E 73 A3 CE 1C AF 10 8E 00 07 EC 81 ED C4 33 C8
$ 7780   DATA   =======   *20 31 3F 26 F5 7A 73 B2 27 01 39 BD 7A 10 12 86
$ 7790   DATA   =======   *FF B7 73 B2 CC 77 B1 FD 73 A1 CC FF FF CE 1C AF
$ 77A0   DATA   =======   *10 8E 00 07 ED C4 33 C8 20 31 3F 26 F7 12 12 12
$ 77B0   DATA   =======   *12 7E 74 C0 00 00 00 00 00 00 00 00 00 00 00 00
$ 77C0   DATA   =======   *FE 71 22 33 C9 FF 7F 86 09 AE A1 AF C4 33 C8 20
$ 77D0   DATA   =======   *4A 26 F6 39 00 00 00 00 00 00 00 00 00 00 00 00
$ 77E0   DATA   =======   *00 12 12 12 8E 06 00 A6 84 88 55 A7 84 30 01 8C
$ 77F0   DATA   =======   *1E 00 25 F3 8E 01 FF 10 8E 10 FF 86 FF CE 00 20
$ 7800   DATA   =======   *C6 10 BD 79 C5 BD 70 60 39 00 00 00 00 00 00 00
$ 7810   DATA   =======   *F7 79 A2 26 05 86 39 B7 70 00 BD 7B 0C 39 00 00
$ 7820   DATA   =======   *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7830   DATA   =======   *00 08 07 12 B6 71 24 27 05 7C 79 21 39 12 BE 78
$ 7840   DATA   =======   *30 30 1F BF 78 30 8C 00 03 23 07 10 8E 78 C0 7E
$ 7850   DATA   =======   *77 C0 8C 00 01 23 07 10 8E 78 CC 7E 77 C0 8C 00
$ 7860   DATA   =======   *02 23 07 10 8E 78 D9 7E 77 C0 8C 00 01 23 07 10
$ 7870   DATA   =======   *8E 78 E8 7E 77 C0 8C 00 02 23 07 10 8E 78 F8 7E
$ 7880   DATA   =======   *77 C0 8C 00 00 27 20 10 BE 71 22 31 21 86 2A A7
$ 7890   DATA   =======   *A4 12 12 12 12 10 8E 79 0A 7E 77 C0 12 12 12 12
$ 78A0   DATA   =======   *12 12 12 12 12 12 12 7C 78 32 BD 77 E4 BD 76 D0
$ 78B0   DATA   =======   *BD 76 FC 7E 71 1B 00 00 00 00 00 00 00 00 00 00
```

```
$ 7C00    DATA    ======    *AA AA AA AA AA AA AA 2A A8 4A AA 2A AA AA AA AA
$ 7C10    DATA    ======    *AA AA A8 0A A8 4A A8 0A AA AA AA AA AA AA 6A A8
$ 7C20    DATA    ======    *0A A4 46 A8 0A AA 6A AA AA AA AA 6A A8 4A A0 02
$ 7C30    DATA    ======    *94 45 A0 02 A8 4A AA 6A AA AA AB 7A AC 4E B0 03
$ 7C40    DATA    ======    *94 45 B0 03 AC 4E AB 7A AA AA AA 6A AF 7E B4 47
$ 7C50    DATA    ======    *B0 03 54 45 B0 03 B4 47 AF 7E AA 6A 00 00 00 00
$ 7C60    DATA    ======    *06 00 12 12 B6 79 21 26 03 7E 78 34 7A 79 20 27
$ 7C70    DATA    ======    *01 39 7A 79 21 86 06 B7 79 20 BE 71 22 33 89 18
$ 7C80    DATA    ======    *00 A6 C4 A7 84 7C 71 20 BD 71 45 10 8E 79 5E F6
$ 7C90    DATA    ======    *71 24 A6 A5 BE 71 22 A7 84 39 12 12 12 12 2A 8A
$ 7CA0    DATA    ======    *A2 A8 00 00 00 86 7A B7 79 45 BD 79 24 86 7C B7
$ 7CB0    DATA    ======    *79 45 39 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7CC0    DATA    ======    *BD 71 02 81 1E 25 01 39 8E 79 24 BF 70 06 8E 79
$ 7CD0    DATA    ======    *65 BF 70 0B BD 71 02 44 8B 46 B7 79 21 7C 79 A1
$ 7CE0    DATA    ======    *39 15 00 00 FC 6F 71 10 83 00 97 25 04 C6 07 20
$ 7CF0    DATA    ======    *0B 54 54 54 54 54 12 CB 03 F0 78 32 7E 78 10 00
$ 7D00    DATA    ======    *03 FF 12 FF FF B7 79 C4 B6 FF 23 8A 38 B7 FF 23
$ 7D10    DATA    ======    *BF 79 C0 10 BF 79 C2 B6 79 C4 B7 FF 20 30 1F 26
$ 7D20    DATA    ======    *F9 BE 79 C0 30 85 BF 79 C0 86 00 B7 FF 20 31 3F
$ 7D30    DATA    ======    *26 F9 10 BE 79 C2 31 A5 10 BF 79 C2 33 5F 11 83
$ 7D40    DATA    ======    *00 00 26 D3 B6 FF 23 84 F7 B7 FF 23 12 12 12 39
$ 7D50    DATA    ======    *C6 01 CE 00 10 86 FF 8E 00 20 10 8E 00 20 7E 79
$ 7D60    DATA    ======    *C5 C6 01 CE 00 10 86 FF 8E 00 80 10 8E 00 80 7E
$ 7D70    DATA    ======    *79 C5 90 FF 7A 7A 32 27 01 39 8E 00 FF BF 7A 32
$ 7D80    DATA    ======    *B6 FF 23 8A 38 B7 FF 23 86 80 B7 FF 20 12 12 12
$ 7D90    DATA    ======    *12 86 FF B7 FF 20 12 12 12 12 B6 FF 23 84 F7 B7
$ 7DA0    DATA    ======    *FF 23 39 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7DB0    DATA    ======    *7A 84 7A 8A 7A 90 7A 96 7A 9C 7A A2 7A A8 7A AE
$ 7DC0    DATA    ======    *7A B4 7A BA A6 99 99 99 99 A6 A6 96 A8 A6 A6 A6
$ 7DD0    DATA    ======    *96 A9 A6 9A 9A 95 96 A9 A6 A9 A9 96 99 99 A5 A9
$ 7DE0    DATA    ======    *A9 A9 95 9A 96 A9 A9 96 A5 9A 95 99 99 A6 95 A9
$ 7DF0    DATA    ======    *A9 A6 A6 A6 A6 99 A6 99 99 A6 A6 99 99 A5 A9 A9
$ 7E00    DATA    ======    *06 7A 00 00 10 8E 06 72 10 BF 7A C0 CE 6F 72 E6
$ 7E10    DATA    ======    *C4 C4 0F BD 7A EE E6 C4 C4 F0 54 54 54 54 BD 7A
$ 7E20    DATA    ======    *EE 33 5F 11 83 6F 6F 26 E6 39 12 12 12 12 58 8E
$ 7E30    DATA    ======    *7A 70 AE 85 86 06 E6 80 12 12 12 12 E7 A4 31 A8
$ 7E40    DATA    ======    *20 4A 26 F2 7A 7A C1 10 BE 7A C0 39 10 8E 06 7C
$ 7E50    DATA    ======    *10 BF 7A C0 CE 79 A2 8E 79 A1 BF 7A E5 BD 7A CF
$ 7E60    DATA    ======    *8E 6F 6F BF 7A E5 39 00 00 00 00 00 00 00 00 00
$ 7E70    DATA    ======    *BD 7B BD 8E 7B 7E 6D 08 27 05 C6 08 F7 70 37 6D
$ 7E80    DATA    ======    *88 10 27 05 C6 05 F7 70 37 6D 88 18 27 05 C6 04
$ 7E90    DATA    ======    *F7 70 37 6D 88 3D 27 03 7C 72 02 6D 04 27 03 7A
$ 7EA0    DATA    ======    *72 02 6D 88 3B 27 03 7A 72 01 6D 02 27 03 7C 72
$ 7EB0    DATA    ======    *01 6D 88 3E 27 03 7C 72 03 7E 72 14 00 00 00 00
$ 7EC0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7ED0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7EE0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7EF0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 BE
$ 7F00    DATA    ======    *FF 00 34 10 BE FF 02 34 10 8E 7B 7E 86 3F 6F 86
$ 7F10    DATA    ======    *4A 2A FB 7F FF 00 7F FF 01 7F FF 03 86 FF B7 FF
$ 7F20    DATA    ======    *02 86 04 B7 FF 01 B7 FF 03 86 08 B7 7B 7C B7 7B
$ 7F30    DATA    ======    *7D 86 FE B7 FF 02 C6 FF B6 FF 00 5C 47 25 02 6C
$ 7F40    DATA    ======    *85 7A 7B 7D 26 F5 7A 7B 7C 27 0C 1A 01 79 FF 02
$ 7F50    DATA    ======    *86 08 B7 7B 7D 20 E1 35 10 BF FF 02 35 10 BF FF
$ 7F60    DATA    ======    *00 39 3F 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7F70    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7F80    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7F90    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7FA0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7FB0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7FC0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7FD0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7FE0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 7FF0    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 8000    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 8010    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 8020    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 8030    DATA    ======    *00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
$ 8040    DATA    ======    *00 F7 FF F7 FF F7 FF F7 FF F7 FF F7 FF F7 FF F7
END OF DATA RUN
```

Concealing Data

***Dene Bebbington** shows you how to encode and decode text files*

THERE ARE times in computing when it is necessary or desirable to hold or transmit data in an encoded form so that unauthorised users cannot examine sensitive or personal data. For example, on the UNIX operating system users' passwords are stored in an encoded form so that even if somebody does get into the password file he will not be able to use them unless he knows the encoding technique and the keyword used. Data encryption methods can to some extent prevent the abuse of computer systems.

There are two main methods of coding, these being substitution and transposition. The substitution methods work by substituting each character by some other, this being determined by the technique and keyword being used. Transposition methods work by changing the order of the characters in the text, so that all the characters are the same but are jumbled up to conceal the original text. Obviously if a technique uses both substitution and transposition then the code will be very hard to break compared to if just one method was used.

Here I present a program which allows users to encode and decode text files (stored on cassette) using a substitution method.

The method presented here is a more advanced version of the PLAYFAIR code which was used in the first world war to encode secret messages. It is based upon a matrix which contains all the available characters, but where the order depends on a keyword entered by the user. Thus as the encoded text depends on the keyword, different text will be obtained from different keywords.

The program (see Listing 1) works by setting up a 7 row by 6 column matrix, thus allowing 42 different characters. The characters that are allowed are "ABCDEFGHIJKLMNOPQRSTUVWXYZ-0123456789*',.;?" (where "*" is a space). A keyword is entered and is stripped of illegal and repeating characters. For example, the keyword "ONCE UPON A TIME." would finally be "ONCE UPATIM.". This is then put into the matrix along with the unused characters. Here the matrix would be:

```
O N C E * U
P A T I M .
B D F G H J
K L Q R S V
W X Y Z 0 1
2 3 4 5 6 7
8 9 ' , ; ?
```

Where again "*" is a space.

Characters are encoded in pairs so that if a line has an odd number of characters a space is added at the end of the line. If the two characters being examined are in the same row then we add 1 to the columns, eg FH would be encoded as GJ. If we go off the end then we 'wrap around', eg LV would be encoded as OK. The same rule applies to characters in the same column except that we add 1 to the rows, so that AX would be encoded as D3. If they are not in the same row or column then we take the characters at the opposite corners, so that BS would be encoded as HK.

Modular

We apply the same principle to decoding except that instead of adding 1 to the row or column we take away 1 from it, remembering to wrap around if necessary. We apply the same rule for characters that are not in the same row or column.

The program works by reading in a text file from cassette to be encoded or decoded, after processing the file is saved back onto cassette in the encoded or decoded form. The user must enter the name of the file and is prompted to prepare the cassette recorder when the file is to be read in or saved.

The program has been written in a modular manner and consists of several subroutines which are called from within the main body of the program, that is lines 2400 to 2680. This makes it easy to modify to make use of a disc drive or allow more characters etc.

The important subroutine is the one at lines 1950 to 2190 as this actually encodes or decodes a message depending on the value of the variable EC, it does this by setting AD to 1 if encoding or −1 if decoding, this is then used to calculate the row or column of pairs of letters.

The other two important subroutines are those to find the position of the two letters in the matrix (lines 1840 to 1890) and to strip illegal characters from a line of text (lines 1090 to 1150).

To set up a text file containing a message to be encoded use the program given in Listing 2. It allows up to 100 lines of text to be entered, and input is terminated when 100 lines have been input or when a full stop is entered on a new line.

As the program needs a keyword to encode and decode you may be wondering which is the best to use. Well, when a method such as this is being used which puts the keyword and remaining letters in a matrix the most effective way is to have a keyword which contains as few repeating characters as possible and contains at least as many characters as is allowed by the program. This makes it more difficult for the code to be broken. For example, the keyword "CONSTANINOPLE, 752914.?" gives a better matrix than the keyword "ZEBEDEE".

It should be remembered that the same keyword must be used for both encoding and decoding otherwise strange results will be obtained!

The ability to conceal data on a computer by encrypting techniques can be useful when the data being held is only to be seen by authorised users or when messages which are being transmitted are to be kept secret. This program could easily be modified to encode data stored in a database and on cassette or disk, or to encode messages sent from computer to computer over a network or telephone line.

Listing 1

```
1 ' ************************
2 ' * DATA ENCODING PROGRAM *
3 ' *                      *
4 ' *  BY: DENE BEBBINGTON  *
5 ' ************************
1000 CLEAR 5000
1010 DIM MX$(7,6),MS$(100)
1020 CR$="ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789 ',.;?"
1030 GOTO 2400
1040 '
1050 ' **********************
1060 ' * STRIP ILLEGAL CHARS *
1070 ' **********************
1080 '
1090 C1=0
1100 FOR C2=1 TO LEN(LN$)
1110 CH$=MID$(LN$,C2,1)
1120 IF INSTR(1,CR$,CH$)<>0 THEN C1=C1+1:MID$(
LN$,C1,1)=CH$
1130 NEXT C2
1140 LL=C1
1150 RETURN
1160 '
1170 ' *******************
1180 ' * READ IN MESSAGE *
1190 ' *******************
1200 '
1210 CLS
1220 PRINT@34,"PREPARE RECORDER FOR INPUT."
1230 PRINT@99,"PRESS ANY KEY WHEN READY."
1240 MOTOR ON
1250 K$=INKEY$:IF K$="" GOTO 1250
1260 PRINT@194,"SEARCHING."
1270 OPEN "I",#-1,NF$
1280 PRINT@194,"READING MESSAGE."
1290 INPUT#-1,NL
1300 FOR CT=1 TO NL
1310 INPUT#-1,MS$(CT)
1320 NEXT CT
1330 CLOSE #-1
1340 RETURN
1350 '
1360 ' *********************
1370 ' * WRITE OUT MESSAGE *
1380 ' *********************
```

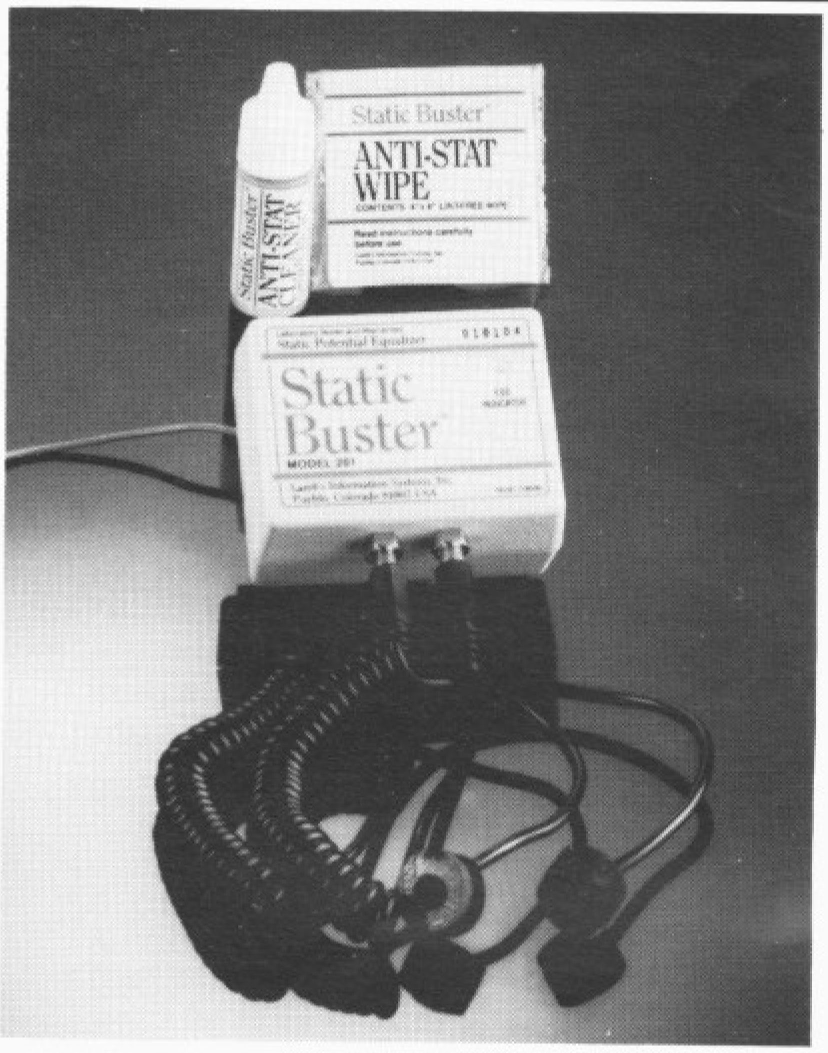
```
1390 '
1400 CLS
1410 PRINT@34,"PREPARE RECORDER FOR OUTPUT."
1420 PRINT@99,"PRESS ANY KEY WHEN READY."
1430 MOTOR ON
1440 K$=INKEY$:IF K$="" GOTO 1440
1450 PRINT@194,"WRITING MESSAGE."
1460 OPEN "O",#-1,NF$
1470 PRINT#-1,NL
1480 FOR CT=1 TO NL
1490 PRINT#-1,MS$(CT)
1500 NEXT CT
1510 CLOSE #-1
1520 RETURN
1530 '
1540 ' ***********************
1550 ' * DROP REPEATING CHARS *
1560 ' ***********************
1570 '
1580 UN$=CR$
1590 KY$=KY$+CR$
1600 C1=0
1610 FOR C2=1 TO LEN(KY$)
1620 CH$=MID$(KY$,C2,1)
1630 PS=INSTR(1,UN$,CH$)
1640 IF PS<>0 THEN C1=C1+1:MID$(KY$,C1,1)=CH$:
MID$(UN$,PS,1)="*"
1650 NEXT C2
1660 RETURN
1670 '
1680 ' ****************
1690 ' * SET UP MATRIX *
1700 ' ****************
1710 '
1720 CT=0
1730 FOR RW=1 TO 7
1740 FOR CL=1 TO 6
1750 CT=CT+1
1760 MX$(RW,CL)=MID$(KY$,CT,1)
1770 NEXT CL,RW
1780 RETURN
1790 '
1800 ' ***************
1810 ' * FIND LETTERS *
1820 ' ***************
1830 '
1840 FOR RW=1 TO 7
1850 FOR CL=1 TO 6
1860 IF L1$=MX$(RW,CL) THEN R1=RW:C1=CL
1870 IF L2$=MX$(RW,CL) THEN R2=RW:C2=CL
1880 NEXT CL,RW
1890 RETURN
1900 '
1910 ' ************************
1920 ' * ENCODE/DECODE MESSAGE *
1930 ' ************************
1940 '
1950 IF EC THEN AD=1 ELSE AD=-1
1960 FOR L1=1 TO NL
1970 LN$=MS$(L1)
1980 LL=LEN(LN$)
1990 GOSUB 1090
2000 IF LL/2<>INT(LL/2) THEN LL=LL+1:LN$=LN$+" "
2010 FOR L2=1 TO LL STEP 2
2020 L1$=MID$(LN$,L2,1)
2030 L2$=MID$(LN$,L2+1,1)
2040 GOSUB 1840
2050 IF R1<>R2 GOTO 2080
2060 C1=C1+AD:IF C1<1 THEN C1=6 ELSE IF C1>6
THEN C1=1
2070 C2=C2+AD:IF C2<1 THEN C2=6 ELSE IF C2>6
THEN C2=1
2080 IF C1<>C2 GOTO 2120
2090 R1=R1+AD:IF R1<1 THEN R1=7 ELSE IF R1>7
THEN R1=1
2100 R2=R2+AD:IF R2<1 THEN R2=7 ELSE IF R2>7
THEN R2=1
2110 GOTO 2130
2120 TP=R1:R1=R2:R2=TP
2130 MID$(LN$,L2,1)=MX$(R1,C1)
2140 MID$(LN$,L2+1,1)=MX$(R2,C2)
2150 NEXT L2
2160 MS$(L1)=LEFT$(LN$,LL)
2170 PRINT MS$(L1)
2180 NEXT L1
2190 RETURN
2200 '
2210 ' ***************
2220 ' * DISPLAY MENU *
2230 ' ***************
2240 '
2250 CLS
2260 PRINT@32,"CHOOSE FROM THE FOLLOWING:-"
2270 PRINT:PRINT
2280 PRINT"1] ENCODE A MESSAGE."
2290 PRINT"2] DECODE A MESSAGE."
2300 PRINT"3] QUIT THE PROGRAM."
2310 PRINT@256,"CHOICE: ";
2320 LINE INPUT CC$
2330 IF CC$<"1" OR CC$>"3" THEN PRINT"MUST
BE 1,2 OR 3.":GOTO 2310
2340 RETURN
2350 '
2360 ' ***************
2370 ' * MAIN PROGRAM *
2380 ' ***************
2390 '
2400 GOSUB 2250
2410 IF CC$="1" THEN EC=1
2420 IF CC$="2" THEN EC=0
2430 IF CC$="3" THEN END
2440 CLS
2450 PRINT@32,"ENTER THE KEYWORD."
2460 PRINT@96,"KEY: ";
2470 LINE INPUT KY$
2480 IF KY$="" GOTO 2460
2490 GOSUB 1580
2500 GOSUB 1720
2510 PRINT@160,"ENTER FILENAME."
2520 PRINT@224,"FILENAME: ";
2530 LINE INPUT NF$
2540 IF NF$="" GOTO 2520
2550 GOSUB 1210
2560 CLS
2570 IF EC THEN PRINT@38,"ENCODING." ELSE
PRINT@38,"DECODING."
2580 PRINT@70,"---------"
2590 PRINT:PRINT
2600 GOSUB 1950
2610 PRINT:PRINT
2620 CLS
2630 PRINT@32,"ENTER FILENAME."
2640 PRINT@96,"FILENAME: ";
2650 LINE INPUT NF$
2660 IF NF$="" GOTO 2640
2670 GOSUB 1400
2680 GOTO 2400
```

Listing 2

```
1 ' *********************
2 ' * TEXT FILE CREATER *
3 ' *********************
1000 CLEAR 5000
1010 DIM MS$(100)
1020 CLS
1030 PRINT"ENTER FILENAME."
1040 PRINT"FILENAME: ";
1050 LINE INPUT NF$
1060 IF NF$="" GOTO 1040
1070 PRINT"ENTER TEXT."
1080 PRINT:PRINT
1090 NL=NL+1
1100 LINE INPUT MS$(NL)
1110 IF NL<>100 AND MS$(NL)<>"."
GOTO 1090
1120 IF MS$(NL)="." THEN NL=NL-1
1130 PRINT:PRINT
1140 PRINT"PREPARE RECORDER"
1150 PRINT"PRESS A KEY WHEN READY"
1160 K$=INKEY$:IF K$="" GOTO 1160
1170 PRINT:PRINT
1180 PRINT"SAVING TEXT FILE."
1190 OPEN "O",#-1,NF$
1200 PRINT#-1,NL
1210 FOR CT=1 TO NL
1220 PRINT#-1,MS$(CT)
1230 NEXT CT
1240 CLOSE#-1
```

News desk

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 4343



Current sponge

INTEGRITY Solutions have produced an anti-static kit for microcomputers. The kit, which comprises two leads with pick-up heads running to a 'static sink' unit, which is in turn attached to an earth lead, costs £49.95 and seems to include an anti-static sponge and cleaning fluid. This might well be of interest to those using expensive computers in a high-static (nylon carpet), low-computeracy environment like an office. Otherwise, static protection can be improvised more cheaply with fine insulated wire run from a VDU screen or micro chassis to a reliable earth point like a steel radiator.

"Recent surveys indicate that some 70% of computer service calls are due to static electricity", state Integrity, and our own electronic consultants confirm that this could, indeed, be the case. Enquiries to Integrity Solutions, 504 Machester Road, Sudden, Rochdale, Lancs OL11 3HE.

Extend the 64

HARRIS Micro Software, previously known for disk-based applications programs for the Dragon 32 and 64, is offering BASIC 42, designed solely for the Dragon 64, which, the makers claim, greatly extends its capabilities.

The program comes on disk, for Dragondos, (though other formats may be available in due course), and makes use of the 64's ability to operate in "Map type 1" with 64K of RAM. The program boots into the cartridge area, above DOS, and thus takes no memory from BASIC. The existing 16K of BASIC, and 8K of DOS, are retained, but modified.

BASIC 42 allows printing on the hi-res screen with standard PRINT commands, using 24 lines of 42 characters per line, redefinable character sets, repeating keys up to eight user-defined windows plus one fixed, but non-destructive window, inverted video, underlining, some extra print commands and functions, commands in lower case, and automatic startup of a BASIC program.

The core program occupies some 4K of memory, and allows other utilities to be loaded in from disk as required. Three such utilities are already written: a program that "patches" the known bugs in Dragondos 1.0, a print "spooler", which allows the computer to be used while the printer is operating, and a "help" utility, which extends some of the features of BASIC 42, with BREAK disable, scroll disables, pause listing, improved TRON (including single-stepping), plus expanded help and error messages.

Harris Micro Software hope to extend the range of utilities to cover extra disk commands, graphics utilities such as icons and sprites, extra sound capabilities, "sidekick" program, communications software, and more. The "open-ended" nature of BASIC 42 means that it is possible to extend the Dragon's capabilities almost indefinitely.

The price for the core program BASIC 42, including alternative character sets, character drawing program, and Dragondos patches, is £14.95. The HELP utility, and the SPOOL utility, will sell for £5 each. The programs will be available from October 1st.

Micro Show

THE Wales and West Computer Show is being held at the Park Hotel, Cardiff on 15th November from 10 to 6.

We have no news as to whether any of the Dragon suppliers will be able to attend, but there should be something to interest multi-micro users.

For more information contact Preston Exhibitions, Kings Hall Court, St. Brdies Major, Mid Glamorgan CF32 0SE. Tel: (0656) 880965.

Classified

MARIDIAN PRESENTS: UNDER-BEINGS OF CROTH: an original adventure set in an underground world of Pitbeasts, toothworms, and apemen. £3.95.
DECATHLON: a 10 game competition for 1-5 players. Includes Shoot-out, Spider, Chopper, Hangman, Anagrams etc. Excellent family entertainment. £3.95
MAZERACE AND SPELLBOX: two interesting approaches to mathematics and spelling for age 6-12. Approved by teachers, parents and children. £3.95 each or £5.95 together.
DRAGON MUSIC: Performances in four part harmony, three pieces per tape. Scott Joplin 1, Bach 1, Bach 2. £3.95 each or £8.95 for all three.
Available only from Maridian, Birchmore Cottage, Nairdwood Lane, Prestwood, Great Missenden, Bucks HP16 OQQ.

Dragon User May '83 to December '86 50p each. Tel: Richard (0784) 254307.

Dragon DOS double diskdrive as new. £150 Telephone 01-302 5758 Evenings only. Lots of Disc Software thrown in.

Dragon Spares — most available. SAE for list including Transformers: 74LS783 £10.25, 6809E £3.25, 6847 £2.50, 2764 £2.10, CPU Boards 32K £22, 64K £38. Nick Spiers, 114 Greenway, Tunbridge Wells Kent TN2 3JN Tel: 0892-44070 Eve, 0622-27404 Day.

Untested. Complete cased dragons, 32K £38, 64K £63, Untested. (Could be working!) All inc p/p Phone Nick 0892 44070.

Dragon 32 Good condition £45 Tel: St Albans 55116. A Turner - Rugg 81 Charmouth Rd St Albans Herts.

Dragon 64 Dragon Disc Amber Printer 12" BWTV Software Books £225 Plant (061) - 485 - 2237. D.A. Plant 10 Ravenswood Drive Cheadle Holme Cheadle, Ches SK8 7D2 Tel: 061-485 2257.

Dragon 32K plus Leads, Recorder, Joysticks, Magazines, Books, over 40 Games, excellent condition £100. Brechin 3407. Mr S.R. McLeod 6 Trinity Fields Crescent.Brechin, Angus DD9 6YF. Tel: Brechin 3407.

Dragon 32. Games, Books Magazines £65 Tel: 0268 754240. Mr T.J. Budd 12 Green Road Benfleet Essex Tel: 0268-754240.

Joystick convertors — 2 Atari — Type Joysticks to any Dragon, £5.95 S.A.E. for details. Cheque1 P.O. to J&S Electronics, 69 Manor Road, Rushden, Northants. NN10 9EX.

Dragon 32, 4 colour printer, tape-deck, joystick interface, 20 games (including shocktroper, Rommel's revenge, +DASM/DEMON cartridge, £120 (O.N.O.). Colin J. Gall 400 Cluny Place, Glenrothes, Fife Tel: 774094 (0592).

Complete modem package for Dragon 32, £80. 0532-694062 ask for Thurain.

Dragon 64, Delta Disk interface, link-wood, editext. Best offer. 031-332 1062.

Modem for sale. 300 baud, leads, software, the lot for £110 o.n.o. Tel: 0437-721 743 weekends only. Ask for Ray.

Dragon 64, Dragon DOS Drive, DMP105 Printer, Joysticks OS9, Basic09, assembler tapes discs books magazines £325 Phone 0436-71036 after 6pm.

Mode 24

***Paul Wicks** raises lower case characters and hi res graphics.*

THIS PROGRAM gives the user true lower case characters on the screen, underlining, inverse text, limited user defined graphics capabilities, 64x192 pixel resolution and still allows all the Dragon's eight colours to be displayed on the screen simultaneously - all from Basic.

The program is written in relocatable machine code, and occupies less than 1k of memory (although it uses the default value of four graphics pages for display).

The hex loader in listing 1 can be used to enter the data in listing 2. The addresses shown in listing 2 place the program at the very end of available RAM although if you wish to leave this memory free of other machine code programs then a lower start address can be used.

To save the program once it is in memory type CSAVEM "filename", start address, start address+&H32F, start address (ENTER) or for the addresses given:
CSAVEM"MODE24",&H7CDO,&H7FFF,&H7CDO (ENTER)

To reload and run the program first type CLEAR 200, start address-1 (ENTER) then CLOADM "filename" (ENTER) then EXEC (ENTER) or with the given addresses:
CLEAR 200,&H7CCF (ENTER)
CLOAD"MODE24" (ENTER)
EXEC (ENTER)

On running the screen clears to amber text on a black background (inverse video, amber screen). BASIC commands, including PRINT@ and CLS n etc. all work as normal with the following additions:

PRINT CHRS (27); reverts to the normal text screen (not mode 24).
PRINT CHRS (28); causes the text to be inverted.
PRINT CHRS (29); switches the underlining on and off.
and PRINT CHRS (30); switches between the amber and green screen display.

These control codes are given in table 2 and an example of how they work is given in listing 6.

(SHIFT) (0) is used as normal to get lower case text except that lower case is now displayed as true lowercase and not as inverse text (to get inverse text use PRINT CHRS (28);"text here"; CHRS (28);...). An example of the type of graphics which can be obtained and of how to set points and draw circles in any colour using BASIC is given in listing 5. To find which character is displayed at a particular point on the screen the functions in listing 4:
DEF FN(P)=PEEK(&H600+352*INT(P/32)-+P)
can be used which returns the PEEK/POKE value (given in table 3) of the character at PRINT@ position N on the screen.

Finally, an example of how to define your own characters for use with MODE24 is given in listing 3 which when used in conjunction with table 1 allows you to redefine any of the lower case characters, CHRS (96) and ASCII characters 123 to 127 in a 5x7 matrix.

Listing 1 — Hex Loader

```
10 'HEXLOADER
20 CLS
30 C=0:CH=0
40 INPUT"START ADDRESS (HEX) ";A$
50 A=VAL("&H"+A$)
60 PRINT"ADDRESS ";HEX$(A);
70 INPUT" BYTE ";P$
80 IF P$="X" THEN END
90 P=VAL("&H"+P$)
100 POKE A,P
110 CH=CH+P
120 C=C+1:IF C=8 GOSUB 140
130 A=A+1:GOTO 60
140 PRINT"CHECKSUM OF LAST 8 BYTES ";HEX$(CH)
150 C=0:CH=0:RETURN
```

Listing 2 — Mode 24 Hex Dump

```
ADDR  HEX DUMP          CHECKSUM>
7CD0 30 8C 5D BF 01 68 30 8D 2FE
7CD8 01 E9 BF 01 A1 30 8D 02 30A
7CE0 23 BF 01 6B 86 7E B7 01 30A
7CE8 67 B7 01 6A B7 01 A0 86 367
7CF0 08 B7 01 8C 86 40 B7 01 2CA
7CF8 95 86 60 B7 01 96 5F BD 3E5
7D00 BA 5F 30 8C 09 A6 80 27 32B
7D08 04 8D 22 20 F8 39 4D 6F 2C0
7D10 64 65 20 32 34 20 20 20 1AF
7D18 28 43 29 20 31 39 38 35 18B
7D20 20 52 2E 50 2E 57 69 63 241
7D28 6B 73 0D 0D 00 8D 01 39 1BF
7D30 0D 6F 27 01 39 32 62 34 1A5
7D38 16 8E FF C0 A7 84 A7 03 438
7D40 A7 05 A7 07 A7 09 A7 0A 2BB
7D48 A7 0C A7 0E A7 88 10 A7 34E
7D50 88 12 B6 01 8C B7 FF 22 3B5
7D58 35 02 34 02 81 08 27 32 14F
7D60 81 0D 27 3E 81 1B 10 27 1C6
7D68 00 86 81 1C 27 6A 81 1D 252
7D70 27 72 81 1E 10 27 00 87 1F6
7D78 81 20 25 5A 81 80 24 0E 253
7D80 81 60 25 04 80 60 20 06 210
7D88 81 40 24 02 8B 40 8D 7B 2BA
7D90 20 1C 86 60 8D 75 30 1E 272
7D98 9F 88 8D 6F 30 1F 9F 88 399
7DA0 20 0C 86 60 8D 65 D6 89 363
7DA8 C4 1F 27 02 20 F6 8C 05 2B3
7DB0 FF 23 23 8E 06 00 EC 89 34E
7DB8 01 80 ED 81 8C 1C 80 25 33C
7DC0 F5 CC 60 60 B8 01 95 F8 4C7
7DC8 01 95 ED 81 8C 1E 00 25 2D3
7DD0 F0 8E 05 E0 9F 88 35 96 455
7DD8 B6 01 95 88 40 B7 01 95 361
7DE0 86 1C 20 F2 B6 01 96 88 389
7DE8 40 B7 01 96 86 1D 20 E6 337
7DF0 86 39 B7 01 67 B7 01 6A 300
7DF8 B7 01 A0 86 1B 20 D7 B6 3A6
7E00 01 8C 88 08 B7 01 8C 86 2E7
7E08 1E 20 CB 34 02 9E 88 1F 284
7E10 10 84 FB C4 E0 30 8B 30 41E
7E18 8B 30 8B 30 8B 30 8B 30 2EC
7E20 8B 30 8B 30 8B 30 8B 30 2EC
7E28 8B 30 8B 30 89 02 00 35 236
7E30 02 81 20 25 46 B8 01 95 25C
7E38 A7 84 A7 88 20 A7 88 40 3E9
7E40 A7 88 60 A7 89 00 80 A7 3E6
7E48 89 00 A0 A7 89 00 C0 A7 3C0
7E50 89 00 E0 A7 89 01 00 A7 341
7E58 89 01 20 A7 89 01 40 A7 2C2
7E60 89 01 60 81 80 24 0A F6 30F
7E68 01 96 F8 01 95 E7 89 01 396
7E70 60 B8 01 95 9E 88 30 01 305
7E78 9F 88 39 34 20 34 02 88 272
7E80 40 B8 01 95 A7 84 A7 88 3E8
7E88 20 A7 88 40 A7 89 01 40 300
7E90 30 88 60 35 02 34 02 C6 24B
7E98 07 3D 31 8D 00 82 31 AB 260
7EA0 C6 07 A6 A0 B8 01 95 A7 408
7EA8 84 30 88 20 5A 26 F3 35 304
7EB0 02 F6 01 96 F8 01 95 E7 404
7EB8 88 20 9E 88 30 01 9F 88 326
7EC0 35 20 39 34 11 AE 63 8C 270
7EC8 BA 62 27 02 35 91 35 11 251
7ED0 27 13 BD 8E 51 C1 08 22 2C1
7ED8 0C 5D 27 06 5A 86 10 3D 1C3
7EE0 CA 0F CA 80 8C C6 60 32 407
7EE8 62 1F 98 34 10 8E 06 00 1F1
7EF0 B8 01 95 F8 01 95 ED 81 44A
7EF8 8C 1E 00 25 F9 8E 04 00 25A
7F00 9F 88 35 90 81 0C 27 03 2A3
7F08 8D 06 39 CC 60 60 8D DB 3C0
7F10 34 16 86 BF 17 FE F4 30 3C8
7F18 1F 9F 88 35 16 39 39 39 23C
7F20 60 4A 75 56 56 63 63 60 2F1
7F28 60 75 4A 7D 55 45 60 4C 2E2
7F30 4C 45 55 55 42 60 60 6A 2A7
7F38 55 4C 55 78 60 4A 4A 6B 2CD
7F40 55 55 45 60 60 6A 55 7D 2EB
7F48 4C 78 60 5E 55 4C 6C 4C 2DB
7F50 46 60 60 7D 55 7D 5C 42 2F3
7F58 60 4C 4C 45 55 55 57 60 29E
7F60 54 60 7B 54 54 78 60 69 318
7F68 60 73 7E 70 70 60 4C 4C 329
7F70 47 4B 70 57 60 7A 54 54 2DB
7F78 54 54 78 60 60 74 4D 40 2E1
7F80 40 57 60 60 75 55 55 55 2CB
7F88 57 60 60 6A 55 55 55 78 2F8
7F90 60 60 6A 44 6A 68 7E 60 31E
7F98 60 6A 70 6A 69 5C 60 60 329
7FA0 75 55 4C 4C 46 60 60 6A 2D2
7FA8 4C 6A 5D 42 60 4C 75 4C 2C2
7FB0 4C 55 78 60 60 55 55 55 2D8
7FB8 55 42 60 60 55 55 55 63 2B9
7FC0 54 60 60 55 55 40 57 57 2AC
7FC8 60 60 55 56 54 63 57 60 2D9
7FD0 60 55 55 7D 5C 78 60 60 31B
7FD8 75 54 7C 4C 45 5E 5E 7D 30F
7FE0 48 57 49 49 63 4D 4D 4F 27D
7FE8 56 56 56 49 5E 7D 48 57 2C5
7FF0 49 49 5E 49 58 4F 56 56 28C
7FF8 56 78 55 7C 45 7C 68 45 30D
```

Listing 3 — User defined graphics

```
10 CLS
20 LET A=96 'ASCII code of character to define.
30 DATA 76,112,77,85,112,84,93
40 ' data to define character (see table 1).
50 GOSUB 100 ' call user define routine.
60 PRINT"This is the character":PRINT"you defined: ";CHR$(A)
70 END
90 '
100 'define character
110 IF A<96 OR A>127 THEN PRINT"CHR$(";A;") OUT OF RANGE":STOP
115 A=INT(A)
120 AD=PEEK(360)*256+PEEK(361)+496+7*(A-96)
130 FOR N=AD TO AD+6
140 READ D
150 POKE N,D
160 NEXT
170 RETURN
```

Listing 4

```
10 DEF FNP(P)=PEEK(&H600+352*INT(P/32)+P)
20 'replaces PEEK(&H400+P)
30 'lower case characters return code of upper case equivalents"
40 '
50 CLS
55 FOR I=0 TO 1
60 FOR N=32 TO 255
70 PRINT@N+I*224,CHR$(N);
80 PRINT@0,USING"£££";FNP(N+I*224);
90 FOR E=1 TO 50:NEXT
100 NEXT
110 PRINTCHR$(28);
120 NEXT
```

Listing 5 — Demonstration of plotting coloured circles in MODE 24

```
10 'CIRCLES ROUTINE
20 'USING MODE 24
30 '
40 CLS0
50 XC=31:YC=95 'centre of circle
60 FOR C=1 TO 8 'colour
70 R=90-8*C 'radius
100 '
110 'circle routine
120 FOR YD=-R TO R
130 XD=SQR(R*R-YD*YD)
140 IF ABS(XD-XL) >3 THEN GOSUB 220
150 X=XC-XD/4:Y=YC-YD:GOSUB 1020
160 X=XC+XD/4:Y=YC-YD:GOSUB 1020
170 XL=XD
180 NEXT
190 NEXT
200 END
210 '
220 FOR XN=XL TO XD STEP 4*SGN(XD-XL)
230 X=XC+XN/4:Y=YC-YD:GOSUB 1020
240 X=XC-XN/4:Y=YC-YD:GOSUB 1020
250 NEXT
260 RETURN
1000 '
1010 'Set X(0-63),Y(0-191),C(0-8)
1020 IF X<0 OR X>63 OR Y<0 OR Y>191 OR C<0 OR C>8 THEN 1110
1030 AD=&H600+INT(X/2)+INT(Y)*32
1040 B=PEEK(AD)
1050 IF B<127 THEN B=128
1060 IF C=0 THEN B=B AND (&HF0+5*(1+(X AND 1))):GOTO 1100
1070 B=B AND &H8F
1080 B=B OR INT(C-1)*16
1090 B=B OR (5*(2-(X AND 1)))
1100 POKE AD,B
1110 RETURN
```

Listing 6 — Demonstration

```
10 CLS
20 U$=CHR$(29)
30 I$=CHR$(28)
40 PRINT"* Upper and lower case text."
50 PRINT
60 PRINT"* ";U$;"Underlining";U$
70 PRINT
80 PRINT"* ";I$;"Inverse video";I$
90 PRINT
100 PRINT"* Special characters: ";CHR$(96);
110 FOR N=123 TO 127:PRINTCHR$(N);:NEXT
120 PRINT:PRINT
130 PRINT"* And all 8 colours ";
140 FOR N=0 TO 7:PRINTCHR$(143+16*N);:NEXT
150 PRINTCHR$(128)
160 PRINT
```

Table 1 — Poke Numbers for user-defined graphics

00000	96	96	96	96	96	96	96
00001	74	74	74	74	93	92	92
00010	124	105	126	0	126	105	124
00011	0	0	0	0	0	0	0
00100	65	84	84	84	84	84	84
00101	0	0	0	0	0	0	0
00110	0	0	0	115	0	0	0
00111	93	0	0	0	0	0	93
01000	126	104	124	0	124	104	126
01001	0	68	68	68	68	68	0
01010	99	99	116	86	86	99	99
01011	0	0	0	0	0	0	0
01100	112	122	122	123	122	122	112
01101	0	0	0	64	0	0	0
01110	120	94	106	120	106	0	120
01111	0	100	0	121	0	0	0
10000	76	76	76	76	76	76	70
10001	85	85	85	85	85	85	87
10010	0	112	112	112	112	112	0
10011	0	0	0	71	0	101	101
10100	0	102	75	0	75	0	0
10101	0	0	77	77	64	64	0
10110	0	0	0	0	0	0	0
10111	0	0	0	0	0	0	0
11000	0	0	0	108	108	0	0
11001	101	101	0	0	0	0	0
11010	0	0	0	0	0	0	0
11011	0	77	99	0	99	87	0
11100	91	0	0	0	0	0	91
11101	0	0	0	0	0	0	0
11110	66	0	117	69	0	100	66
11111	69	0	125	107	125	0	69

Table 2 — control codes

CHR$(27) - Return to normal (not mode24) text screen

CHR$(28) - Inverse / true video toggle

CHR$(29) - Underline on/off toggle

CHR$(30) - Green / amber screen toggle

Table 3 — code for FN(P)

CHR$	CHR	normal	inverse
32		96	32
33	!	97	33
34	"	98	34
35	#	99	35
36	$	100	36
37	%	101	37
38	&	102	38
39	'	103	39
40	(	104	40
41	)	105	41
42	*	106	42
43	+	107	43
44	,	108	44
45	-	109	45
46	.	110	46
47	/	111	47
48	0	112	48
49	1	113	49

continued

				74	J	74	10	101	e	69	5				
				75	K	75	11	102	f	70	6				
CHR$	CHR	normal	inverse	76	L	76	12	103	g	71	7				
50	2	114	50	77	M	77	13	104	h	72	8				
51	3	115	51	78	N	78	14	105	i	73	9				
52	4	116	52	79	O	79	15	106	j	74	10				
53	5	117	53	80	P	80	16	107	k	75	11				
54	6	118	54	81	Q	81	17	108	l	76	12				
55	7	119	55	82	R	82	18	109	m	77	13				
56	8	120	56	83	S	83	19	110	n	78	14				
57	9	121	57	84	T	84	20	111	o	79	15				
58	:	122	58	85	U	85	21	112	p	80	16				
59	;	123	59	86	V	86	22	113	q	81	17				
60	<	124	60	87	W	87	23	114	r	82	18				
61	=	125	61	88	X	88	24	115	s	83	19				
62	>	126	62	89	Y	89	25	116	t	84	20				
63	?	127	63	90	Z	90	26	117	u	85	21				
64	@	64	0	91	[	91	27	118	v	86	22				
65	A	65	1	92	\	92	28	119	w	87	23				
66	B	66	2	93	]	93	29	120	x	88	24				
67	C	67	3	94	↑	94	30	121	y	89	25				
68	D	68	4	95	←	95	31	122	z	90	26				
69	E	69	5	96	π	64	0	123	♠	91	27				
70	F	70	6	97	a	65	1	124	♡	92	28				
71	G	71	7	98	b	66	2	125	♣	93	29				
72	H	72	8	99	c	67	3	126	♢	94	30				
73	I	73	9	100	d	68	4	127	£	95	31				

```
                                      NAM   MODE24 ASSEMBLY LISTING
0195                        INVERT    EQU   $195
0196                        ULINE     EQU   $196
018C                        MODE      EQU   $18C
7CD0                                  ORG   $7CD0
7CD0  30 8C 5D      (7D30)  INIT      LEAX  START,PCR
7CD3  BF 0168                         STX   360
7CD6  30 8D 01E9    (7EC3)            LEAX  >CLS,PCR
7CDA  BF 01A1                         STX   417
7CDD  30 8D 0223    (7F04)            LEAX  >KEY,PCR
7CE1  BF 016D                         STX   363
7CE4  86 7E                           LDA   £126
7CE6  B7 0167                         STA   359
7CE9  B7 016A                         STA   362
7CEC  B7 01A0                         STA   416
7CEF  86 08                           LDA   £8
7CF1  B7 018C                         STA   MODE
7CF4  86 40                           LDA   £64
7CF6  B7 0195                         STA   INVERT
7CF9  86 60                           LDA   £96
7CFB  B7 0196                         STA   ULINE
7CFE  5F                              CLRB
7CFF  BD BA5F                         JSR   $BA5F
7D02  30 8C 09      (7D0E)            LEAX  MESSAGE,PCR
7D05  A6 80                 :2        LDA   ,X+
7D07  27 04         (7D0D)            BEQ   :1
7D09  8D 22         (7D2D)            BSR   OUT
7D0B  20 F8         (7D05)            BRA   :2
7D0D  39                    :1        RTS
7D0E 4D6F64652032           MESSAGE   FCC   /Mode 24   (C) 1985 R.P.Wicks/
7D2A 0D0D00                           FCB   13,13,0
7D2D  8D 01         (7D30)  OUT       BSR   START
7D2F  39                              RTS

7D30  0D 6F                 START     TST   111
7D32  27 01         (7D35)            BEQ   SCREEN
7D34  39                              RTS
7D35  32 62                 SCREEN    LEAS  2,S
7D37  34 16                           PSHS  A,B,X
7D39  8E FFC0                         LDX   £$FFC0
7D3C  A7 84                           STA   ,X
7D3E  A7 03                           STA   3,X
7D40  A7 05                           STA   5,X
7D42  A7 07                           STA   7,X
7D44  A7 09                           STA   9,X
7D46  A7 0A                           STA   10,X
7D48  A7 0C                           STA   12,X
7D4A  A7 0E                           STA   14,X
7D4C  A7 88 10                        STA   16,X
7D4F  A7 88 12                        STA   18,X
7D52  B6 018C                         LDA   MODE
7D55  B7 FF22                         STA   $FF22
7D58  35 02                           PULS  A
7D5A  34 02                           PSHS  A
7D5C  81 08                           CMPA  £8
7D5E  27 32         (7D92)            BEQ   BSPACE
7D60  81 0D                           CMPA  £13
7D62  27 3E         (7DA2)            BEQ   LFEED
7D64  81 1B                           CMPA  £27
7D66 1027 0086      (7DF0)            LBEQ  ESC
7D6A  81 1C                           CMPA  £28
7D6C  27 6A         (7DD8)            BEQ   INVERSE
7D6E  81 1D                           CMPA  £29
7D70  27 72         (7DE4)            BEQ   UNDER
7D72  81 1E                           CMPA  £30
7D74 1027 0087      (7DFF)            LBEQ  GREEN
7D78  81 20                           CMPA  £32
7D7A  25 5A         (7DD6)            BLO   NOSCROLL
7D7C  81 80                           CMPA  £128
7D7E  24 0E         (7D8E)            BHS   :1
7D80  81 60                           CMPA  £96
7D82  25 04         (7D88)            BLO   :2
7D84  80 60                           SUBA  £96
7D86  20 06         (7D8E)            BRA   :1
7D88  81 40                 :2        CMPA  £64
7D8A  24 02         (7D8E)            BHS   :1
7D8C  8B 40                           ADDA  £64
7D8E  8D 7B         (7E0B)  :1        BSR   PRINT
7D90  20 1C         (7DAE)            BRA   DONE
7D92  86 60                 BSPACE    LDA   £96
7D94  8D 75         (7E0B)            BSR   PRINT
7D96  30 1E                           LEAX  -2,X
7D98  9F 88                           STX   136
7D9A  8D 6F         (7E0B)            BSR   PRINT
7D9C  30 1F                           LEAX  -1,X
7D9E  9F 88                           STX   136
7DA0  20 0C         (7DAE)            BRA   DONE
7DA2  86 60                 LFEED     LDA   £96
7DA4  8D 65         (7E0B)  :1        BSR   PRINT
7DA6  D6 89                           LDB   137
7DA8  C4 1F                           ANDB  £$1F
7DAA  27 02         (7DAE)            BEQ   DONE
7DAC  20 F6         (7DA4)            BRA   :1
7DAE  8C 05FF               DONE      CMPX  £1535
7DB1  23 23         (7DD6)            BLS   NOSCROLL
7DB3  8E 0600                         LDX   £$600
7DB6  EC 89 0180            SCROLL    LDD   32*12,X
7DBA  ED 81                           STD   ,X++
7DBC  8C 1C80                         CMPX  £-32*12+$1E00
7DBF  25 F5         (7DB6)            BLO   SCROLL
7DC1  CC 6060               :1        LDD   £96*256+96
7DC4  B8 0195                         EORA  INVERT
7DC7  F8 0195                         EORB  INVERT
7DCA  ED 81                           STD   ,X++
7DCC  8C 1E00                         CMPX  £$1E00
7DCF  25 F0         (7DC1)            BLO   :1
7DD1  8E 05E0                         LDX   £1536-32
7DD4  9F 88                           STX   136
```

```
7DD6   35 96              NOSCROLL PULS  A,B,X,PC
7DD8   B6 0195            INVERSE  LDA   INVERT
7DDB   88 40                       EORA  £64
7DDD   B7 0195                     STA   INVERT
7DE0   86 1C                       LDA   £28
7DE2   20 F2      (7DD6)           BRA   NOSCROLL
7DE4   B6 0196            UNDER    LDA   ULINE
7DE7   88 40                       EORA  £64
7DE9   B7 0196                     STA   ULINE
7DEC   86 1D                       LDA   £29
7DEE   20 E6      (7DD6)           BRA   NOSCROLL
7DF0   86 39              ESC      LDA   £$39
7DF2   B7 0167                     STA   359
7DF5   B7 016A                     STA   362
7DF8   B7 01A0                     STA   416
7DFB   86 1B                       LDA   £27
7DFD   20 D7      (7DD6)           BRA   NOSCROLL
7DFF   B6 018C            GREEN    LDA   MODE
7E02   88 08                       EORA  £8
7E04   B7 018C                     STA   MODE
7E07   86 1E                       LDA   £30
7E09   20 CB      (7DD6)           BRA   NOSCROLL

7E0B   34 02              PRINT    PSHS  A
7E0D   9E 88                       LDX   136
7E0F   1F 10                       TFR   X,D
7E11   84 FB                       ANDA  £$FB
7E13   C4 E0                       ANDB  £$E0
7E15   30 8B                       LEAX  D,X
7E17   30 8B                       LEAX  D,X
7E19   30 8B                       LEAX  D,X
7E1B   30 8B                       LEAX  D,X
7E1D   30 8B                       LEAX  D,X
7E1F   30 8B                       LEAX  D,X
7E21   30 8B                       LEAX  D,X
7E23   30 8B                       LEAX  D,X
7E25   30 8B                       LEAX  D,X
7E27   30 8B                       LEAX  D,X
7E29   30 8B                       LEAX  D,X
7E2B   30 89 0200                  LEAX  $200,X
7E2F   35 02                       PULS  A
7E31   81 20                       CMPA  £32
7E33   25 46      (7E7B)           BLO   LCASE
7E35   B8 0195                     EORA  INVERT
7E38   A7 84                       STA   ,X
7E3A   A7 88 20                    STA   32,X
7E3D   A7 88 40                    STA   64,X
7E40   A7 88 60                    STA   96,X
7E43   A7 89 0080                  STA   128,X
7E47   A7 89 00A0                  STA   160,X
7E4B   A7 89 00C0                  STA   192,X
7E4F   A7 89 00E0                  STA   32*7,X
7E53   A7 89 0100                  STA   32*8,X
7E57   A7 89 0120                  STA   32*9,X
7E5B   A7 89 0140                  STA   320,X
7E5F   A7 89 0160                  STA   32*11,X
7E63   81 80                       CMPA  £128
7E65   24 0A      (7E71)           BHS   :3
7E67   F6 0196                     LDB   ULINE
7E6A   F8 0195                     EORB  INVERT
7E6D   E7 89 0160                  STB   32*11,X
7E71   B8 0195            :3       EORA  INVERT
7E74   9E 88                       LDX   136
7E76   30 01                       LEAX  1,X
7E78   9F 88                       STX   136
7E7A   39                          RTS
7E7B   34 20              LCASE    PSHS  Y
7E7D   34 02                       PSHS  A
7E7F   88 40                       EORA  £64
7E81   B8 0195                     EORA  INVERT
7E84   A7 84                       STA   ,X
7E86   A7 88 20                    STA   32,X
7E89   A7 88 40                    STA   64,X
7E8C   A7 89 0140                  STA   320,X
7E90   30 88 60                    LEAX  96,X
7E93   35 02                       PULS  A
7E95   34 02                       PSHS  A
7E97   C6 07                       LDB   £7
7E99   3D                          MUL
7E9A   31 8D 0082 (7F20)           LEAY  >TABLE,PCR
7E9E   31 AB                       LEAY  D,Y
7EA0   C6 07                       LDB   £7
7EA2   A6 A0              :1       LDA   ,Y+
7EA4   B8 0195                     EORA  INVERT
7EA7   A7 84                       STA   ,X
7EA9   30 88 20                    LEAX  32,X
7EAC   5A                          DECB
7EAD   26 F3      (7EA2)           BNE   :1
7EAF   35 02                       PULS  A
7EB1   F6 0196                     LDB   ULINE
7EB4   F8 0195                     EORB  INVERT
7EB7   E7 88 20                    STB   32,X
7EBA   9E 88                       LDX   136
7EBC   30 01                       LEAX  1,X
7EBE   9F 88                       STX   136
7EC0   35 20                       PULS  Y
7EC2   39                          RTS

7EC3   34 11              CLS      PSHS  CC,X
7EC5   AE 63                       LDX   3,S
7EC7   8C BA62                     CMPX  £47714
7ECA   27 02      (7ECE)           BEQ   CONT
7ECC   35 91                       PULS  CC,X,PC
7ECE   35 11              CONT     PULS  CC,X
7ED0   27 13      (7EE5)           BEQ   :1
7ED2   BD 8E51                     JSR   $8E51
7ED5   C1 08                       CMPB  £8
7ED7   22 0C      (7EE5)           BHI   :1
7ED9   5D                          TSTB
7EDA   27 06      (7EE2)           BEQ   :2
7EDC   5A                          DECB
7EDD   86 10                       LDA   £$10
7EDF   3D                          MUL
7EE0   CA 0F                       ORB   £$0F
7EE2   CA 80              :2       ORB   £$80
7EE4 8C                            FCB   $8C
7EE5   C6 60              :1       LDB   £$60
7EE7   32 62              CLR      LEAS  2,S
7EE9   1F 98                       TFR   B,A
7EEB   34 10              FILL     PSHS  X
7EED   8E 0600                     LDX   £$600
7EF0   B8 0195                     EORA  INVERT
7EF3   F8 0195                     EORB  INVERT
7EF6   ED 81              :1       STD   ,X++
7EF8   8C 1E00                     CMPX  £$1E00
7EFB   25 F9      (7EF6)           BLO   :1
7EFD   8E 0400                     LDX   £$400
7F00   9F 88                       STX   136
7F02   35 90                       PULS  X,PC
7F04   81 0C              KEY      CMPA  £12
7F06   27 03      (7F0B)           BEQ   :1
7F08   8D 06      (7F10)           BSR   CURSOR
7F0A   39                          RTS
7F0B   CC 6060            :1       LDD   £$6060
7F0E   8D DB      (7EEB)           BSR   FILL
7F10   34 16              CURSOR   PSHS  A,B,X
7F12   86 BF                       LDA   £143+48
7F14   17 FEF4    (7E0B)           LBSR  PRINT
7F17   30 1F                       LEAX  -1,X
7F19   9F 88                       STX   136
7F1B   35 16                       PULS  A,B,X
7F1D   39                          RTS
7F1E   39                          RTS
7F1F   39                          RTS


                          * CHARACTER SET
7F20 604A75565663         TABLE    FCB   96,74,117,86,86,99,99
7F27 6060754A7D55                  FCB   96,96,117,74,125,85,69
7F2E 604C4C455555                  FCB   96,76,76,69,85,85,66
7F35 60606A554C55                  FCB   96,96,106,85,76,85,120
7F3C 604A4A6B5555                  FCB   96,74,74,107,85,85,69
7F43 60606A557D4C                  FCB   96,96,106,85,125,76,120
7F4A 605E554C6C4C                  FCB   96,94,85,76,108,76,70
7F51 60607D557D5C                  FCB   96,96,125,85,125,92,66
7F58 604C4C455555                  FCB   96,76,76,69,85,85,87
7F5F 6054607B5454                  FCB   96,84,96,123,84,84,120
7F66 606960737E70                  FCB   96,105,96,115,126,112,112
7F6D 604C4C474B70                  FCB   96,76,76,71,75,112,87
7F74 607A54545454                  FCB   96,122,84,84,84,84,120
7F7B 6060744D4040                  FCB   96,96,116,77,64,64,87
7F82 606075555555                  FCB   96,96,117,85,85,85,87
7F89 60606A555555                  FCB   96,96,106,85,85,85,120
7F90 60606A446A68                  FCB   96,96,106,68,106,104,126
7F97 60606A706A69                  FCB   96,96,106,112,106,105,92
7F9E 606075554C4C                  FCB   96,96,117,85,76,76,70
7FAC 604C754C4C55                  FCB   96,76,117,76,76,85,120
7FB3 606055555555                  FCB   96,96,85,85,85,85,66
7FBA 606055555563                  FCB   96,96,85,85,85,99,84
7FC1 606055554057                  FCB   96,96,85,85,64,87,87
7FC8 606055565463                  FCB   96,96,85,86,84,99,87
7FCF 60605555 7D5C                 FCB   96,96,85,85,125,92,120
7FD6 606075547C4C                  FCB   96,96,117,84,124,76,69
7FDD 5E5E7D485749                  FCB   94,94,125,72,87,73,73
7FE4 634D4D4F5656                  FCB   99,77,77,79,86,86,86
7FEB 495E7D485749                  FCB   73,94,125,72,87,73,73
7FF2 5E49584F5656                  FCB   94,73,88,79,86,86,86
7FF9 78557C457C68                  FCB   120,85,124,69,124,104,69
```

Quick CLS

***A Daniel** clears semigraphics the fast way with machine code.*

THIS LITTLE programme is a simple machine code routine that can quickly clear any part of the low or high resolution screens or even the semigraphics screen which can only be cleared very slowly in Basic.

It is really a 'must' for all basic users and no knowledge of machine code is required except, as with all machine code programmes you must save it before you try it as a typing mistake may cause an irretrievable crash.

You do, however, need to know the memory addresses of the screen locations. To remind you, the low resolution screen starts at 1024 and continues through 511 locations to 1535. The High Res. screen starts at 1536 and uses a variable number of pages according to the mode (see Manual), each page containing 1536 locations. The semigraphics 24 screen starts at 1024 and ends at 7679.

The listing below gives a free running demonstration of blocks of colour being placed on areas of high res. and low res. screens on which a much slower basic pattern has been drawn. The semigraphics 24 screen is then cleared completely and a blue squiggle added to prove that it really is semigraphics! However I hope you will have realized by now that the potential of this routine is enormous.

To incorporate the first half of the listing in your own programmes you must give values to 3 variables — S1, S2 and S3 as in the REM line 30. S1 is the start address of the area to be cleared and must not be less than 1024. S2 is the end address of the area and S3 is the clearing colour. Of course you must also set up the mode and screen for high res. graphics or poke the appropriate numbers (as in line 390) for semigraphics.

To produce a uniform colour in low res. or semigraphics you can just enter the normal number (0-8) of the colour you require as the variable S3. In high res. any number between 0 and 255 will produce screen patterns but only 6 numbers will give uniform shades. These are 0 which produces the normal background colour, 255 - the normal foreground, and 85 and 170 which give the two remaining colours. The other two numbers 187 and 238 produce a composite colour which on my television set will pass as a useful fifth shade! When using high res. (or if you wish to make patterns on low res.) you must delete lines 90 and 100 as otherwise the value of S3 will be corrupted.

The machine code pokes are contained in a subroutine so the programme proper ends at line 50. You can go from here to wherever you want and to operate the clearing just write EXEC30000.

If your programme needs a lot of string space increase the 200 in line 10. If you are using other machine code routines you may wish to lower the 29999 figure in line 10 or alter the start address in line 60.

```
   30 REM INPUT"ENTER START POINT, END P
OINT, COLOUR";S1,S2,S3
   40 GOSUB60
   50 GOTO220
   60 A=30000
   70 IF D=1 THEN 110
   80 IF S3<1 THEN S3=1 ELSE S3=S3*16
   90 IF S3>128 THEN S3=128: REM THIS LI
NE AND LINE 100 NOT REQUIRED IN HIGH RES
. MODES. VALUES FOR S3 CAN BE FOUND IN L
INE 230
  100 S3=S3+127
  110 POKEA,&H8E
  120 IF S1<S2 AND S1>1023 AND S2>1023 A
ND S1<7681 AND S2<7681 THEN GOSUB180 ELS
E POKEA+1,&H04:POKEA+2,&H00: POKEA+8,&H0
6: POKEA+9,&H00
  130 POKEA+3,&H86
  140 POKEA+4,S3
  150 POKEA+5,&HA7: POKEA+6,&H80: POKEA+
7,&H8C
  160 POKEA+10,&H25: POKEA+11,&HF9: POKE
A+12,&H39
  170 RETURN
  180 S5=INT(S1/256): S6=S1-S5*256
  190 S7=INT(S2/256): S8=S2-S7*256
  200 POKEA+1,S5: POKEA+2,S6: POKEA+8,S7
:POKEA+9,S8
  210 RETURN
  220 REM THIS IS A DEMONSTRATION ONLY:
TO CONSTRUCT YOUR OWN PROGRAMME YOU CAN
OMIT ALL THE FOLLOWING LINES
  230 S4(0)=0:S4(1)=85:S4(2)=170: S4(3)=
187: S4(4)=238: S4(5)=255
  240 D=D+1: ON D GOSUB340,310,390
  250 GOSUB60
  260 EXEC30000
  270 IF D=3 THEN FOR J=1 TO 1000: NEXT:
 FOR J=3000 TO 6500 STEP 33: POKEJ,175:
NEXT
  280 FOR J=1 TO 1000: NEXT
  290 IF D=3 THEN D=0:PRINT@100,"THAT WA
S SEMIGRAPHICS 24";: FOR J=1 TO 1000: NE
XT
  300 GOTO240
  310 CLS:FOR B=200 TO 329: PRINT@B,CHR$
(RND(225)+30);: NEXT B
  320 S1=1184: S2=1536: S3=RND(9)-1
  330 RETURN
  340 PMODE3,1: SCREEN1,0: PCLS
  350 FOR B=1 TO 15: CIRCLE(RND(150)+50,
RND(100)+50),RND(40),RND(4): NEXT
  360 S1=1536: S2=5000
  370 S3=S4(S9): S9=S9+1: IF S9>5 THEN S
9=0
  380 RETURN
  390 POKE65472,0: POKE65475,1: POKE6547
7,1
  400 FOR J=1 TO 1000: NEXT
  410 S1=1024: S2=7679: S3=RND(9)-1
  420 RETURN
```

Expert's Arcade Arena

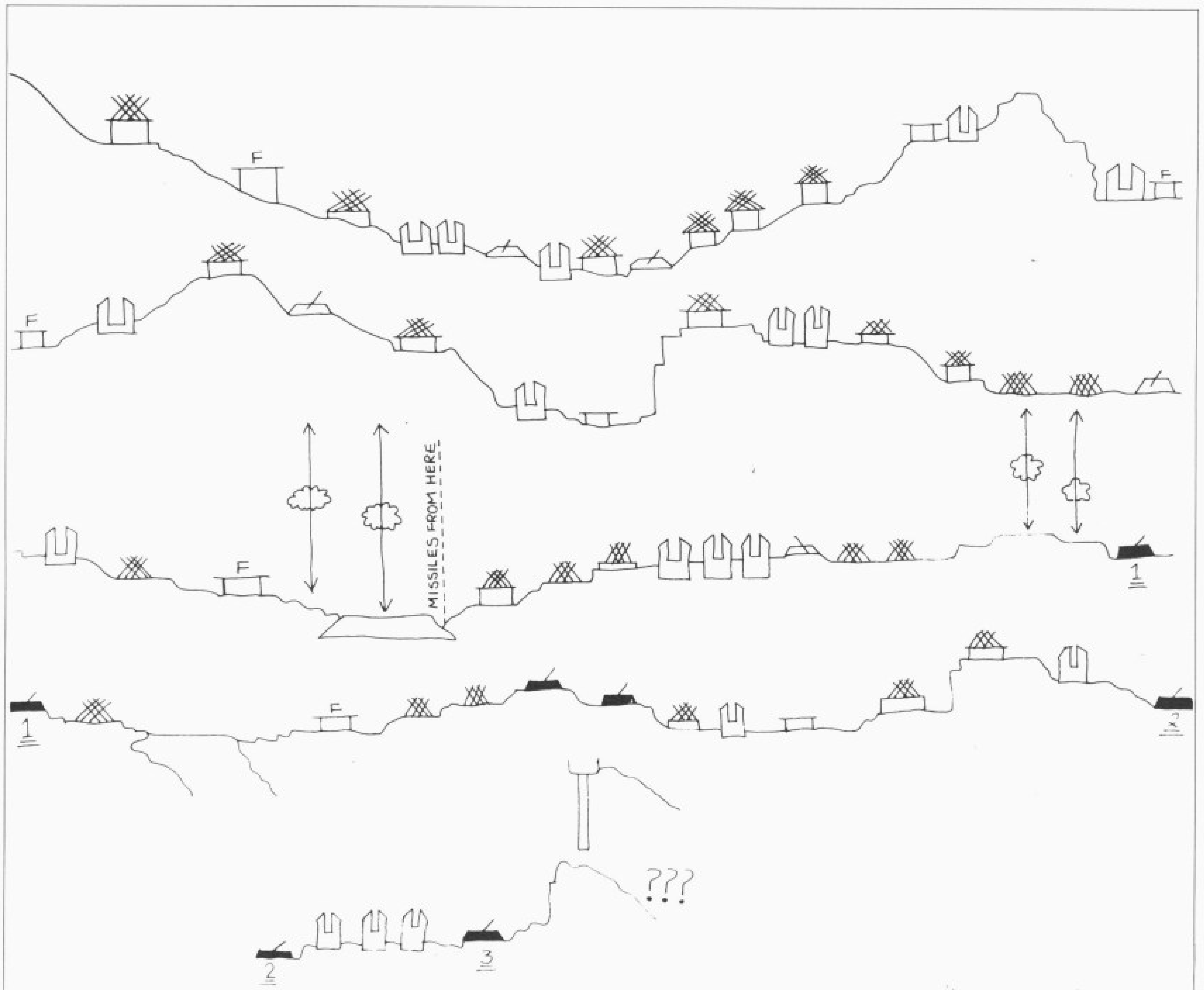
Write to 'The Expert' at Dragon User 12-13 Little Newport St, London WC2H 7PP. with all your arcade tips and hints.

JUST a shortish column this month because of its accompaniment, yes, for those of you newly acquainted with the term, masterpiece, here's a map of the opening section of *Copta Snatch*, together with some VERY helpful advice. To aid and enhance chances of getting past that line that goes up and down ONLY THREE GUNS need be taken out. They are marked on the Map as 1, 2, and 3 for pretty obvious reasons. Gun one must be taken out to enable a 100% success rate on take off if one should die while trying to take out guns two and three. These are the guns that hinder you while trying to get past the up and down thingy!! Right, that's the diagram explained, now then, who'd care to send me a map of the second stage. Plus a few tips on how to get there!!

The winner of the "Who is The Expert?" Competition this month is a certain Andrew McLachlan from Woking in Surrey, who suggests that I am in fact Brian of Nazareth. Nice try Andy, wrong, but a nice try, and your prizes number three, two abstract and one material. Firstly, two pieces of totally worthless but sensible pieces of advice... 1) Keep away from girls with spiky hair! and 2) Stop watching so much Monty Python. And your material Prize, which you should have received about a month ago, is a Photograph, taken by my own fair hand, of a clump of grass from my back garden. I know, words fail you, you've been wondering who sent you that Photo for about a month haven't you?

Now then, some games news... Smithson are producing a game called Frankie about which I shall reveal no more but looks like being very good indeed, keep reading this column for more news... New release from Microdeal soon looks like being a version of BoulderCrash, with 28 screens (I think), a randomize choice (a-la *Screaming Abdabs*), and some wicked screens involving slime; this one looks like a must!!

Your letters are still arriving, by the crate load, oh yes, and by the way, this luminous pen craze is getting beyond a joke!! This month I have received two letters in Luminous Pen from a Helen Marsdon in Camberley, and an Andrew Ridgely in Wales (surely not the Andrew Ridgely) anyway Andrew Wize Guy Yellow Pen Ridgely and Helen Nymphomaniac (you should read her letter — I'll send a copy to the first person who... no, what am I thinking about!) Pink Pen Marsdon, I've informed Norman Tebbit of your addresses and he'll be round for your conservative party membership fees very soon... Heh Heh Heh. Anyway, your letters. A cry from one Peter Kertin from Lancashire, "Dear Expert, please provide a map for at least the start of *Copta-Snatch* as I think this would be a great help to many people" — And they say I never answer requests! —"and also have you got around to maps for *Fantasy Fight* and *Brock's Kingdom* yet?" — These I leave to you dear readership! — "Please, same help with *Gis A Job*" — I'm working on it! — "Finally, to say that I love your column and that I find it is the part of the magazine I turn to first... You should be allocated a lot more space! Do you write for any other magazines?" — No! — "Please, keep up the great work!" Thank you and good night!!



If you've got a technical question write to Brian Cadge. Please do not send a SAE as Brian cannot guarantee to answer individual inquiries.

Dragon Answers

Baud rate

I AM thinking of buying a modem, but want to know if I would need a Dragon 64 as I only have a 32 at the moment. Could you explain baud rate and start/stop bits when used with modems?

S.J. Scoltoch
17 Oakdale Rd
Downend
Bristol

MOST modems talk to the computer via an RS232 serial interface which is built in on the 64 but not on the 32. Therefore, you either need to get a new Dragon 64 or buy one of the many RS232 cartridges available for your 32. Some company's such as Compusense and Peaksoft will sell you an all in one package consisting of modem, RS232 cartridge and software ready to run.

The 'baud rate' is the speed at which data is sent down the serial interface. It is actually the number of times the signal line changes state in a second, but is often simply referred to as the number of bits sent per second.

Start bits are special bits sent before a byte of data to 'wake up' the computer at the other end and allow it to syncronize to the incoming data, stop bits simply insert a gap between bytes of data and are sent after a byte of data.

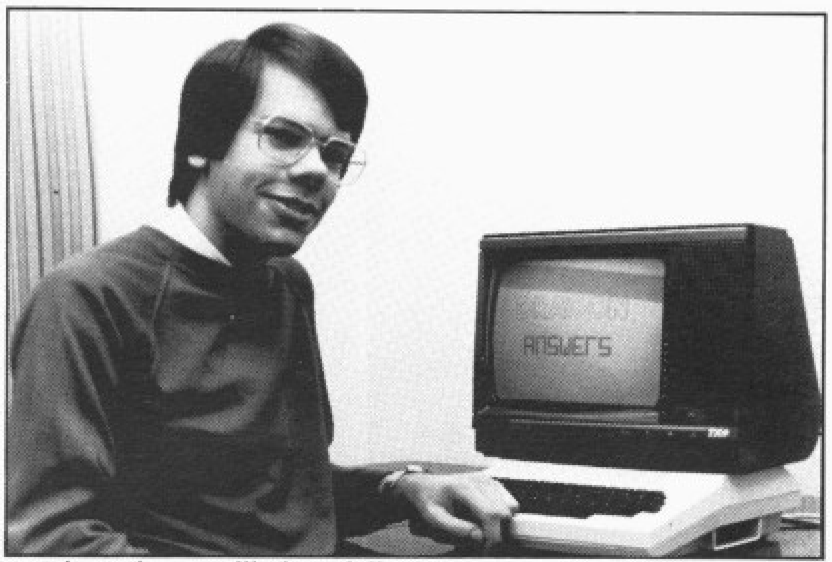
Keyboard

I HAVE a Dragon 32 and my problem is that 3 of the keys do not work. They are the CLEAR, ENTER and SHIFT keys. Can you tell me what's wrong and how to mend it?

S. Hardy
10 Crescent
Mathern
Chepstow
Gwent NP6 6JN

YOUR problem is almost certainly caused by a failing keyboard unit, and not by the CPU or other components. The 3 keys you mention (plus the BREAK key which you'll probably find doesn't work either) are all on the same input line from the keyboard matrix and so are likely to fail together. This means that the keyboard 'keys' are probably fine and there is simply a loose connection between keyboard and ribbon cable or ribbon cable and CPU board.

Alternatively, try contacting one of the repair services found on the pages of Dragon User for a quote on a new keyboard — you should be able to fit this yourself.



Music

I AM in the process of writing a game on my Dragon 64 and would like to know if it is possible to produce "Interrupt Driven Music". I know this is possible on the Commodore, Amstrad etc., but they have a different hardware.

Rolf Michelsen
Steinkjer
Norway

THE DRAGON is certainly not famous for its tremendous sound capabilities, this is due to the fact that unlike almost every other decent home computer it does not contain a dedicated sound chip. Hence, when the Dragon wants to make a noise the 6809 must do all the work and cannot do anything else.

Therefore, it is not possible to produce true 'Interrupt' sound. However, one fudge used in some games (like Microdeal's "Crazy Painter") is to have a routine which produces very short 'beeps' and call this routine from within the main game loop. You're limited to up tempo music but the results can be quite effective.

Sequel

THE OTHER day I was playing a game called 'The Ring of Darkness' on a friend's machine. He said he had had the game for a couple of years, now I am interested in buying the game but I can not find where to buy it. Could you tell me who produce the game and where I can get it from?

Bryn Allison
27 St. Marys Park
Louth
Lincs LN11 0EF

THE RING of Darkness was produced by Wintersoft who have since released the sequel called 'Return of the Ring'. Both can still be obtained direct from Wintersoft at 30 Uplands Park Road, Enfield, Middlesex, priced at £6.95 each.

Stacked

I AM in the process of writing an assembler (in Basic) for my Dragon 32. The program is menu driven and I want a 'status' mode available, where the status of the flags register is shown.

The problem is where is the 'CC' stored? My reference manual says it is stored on the stack, but any attempt to pull the CC will result in a different answer.

Martyn Smith
43 Church Road
Cowley
Uxbridge

I GATHER from your problems regarding the 'CC' (condition codes) register that you are actually writing a dis assembler rather than an assembler. The CC register is internal to the 6809 and is not 'stored' in memory at any physical address. You can access it in machine code by pushing it onto the stack with a PSHS CC command and getting it back into the accumulator with a PULS A command, followed by STA to store it in memory.

Of course, the CC is not relevant to a disassembler anyway and would only be referred to by a 6809 simulator which would be rather difficult to run in BASIC.

Sorting

I HAVE written a Mailing Address program for my Dragon with disc drive. It has many functions such as word search and list all addresses of same type etc. The problem I have is with sorting. The only way I know of sorting a file is to load it all into memory, sort it and rewrite to disc. This limits the number of records to around 200. Is there a way of being able to sort the file without having this limitation?

Robin Smart
40 Chaucer House
Chaucer Gardens
Sutton
Surrey

THE solution to your problem seems to be to use random access files rather than load all the data into memory at once. You'll need to use the FREAD and FWRITE commands and decide on a fixed record length (128 characters should cover the longest name and address). Then a particular record 'N' in the file can be retrieved into A$ using;

```
FLREAD "FILENAME",AT
   (N-1)*128,
   FOR 128; A$
```

Most sort routines include lines to compare two strings and, if necessary, swap the two strings. All you need to do is read in the correct records from disc before the comparison is made and write them back out if they are swapped over (you could use the SWAP command for this).



ROLL UP, roll up, roll up, get your free solution here folks, guaranteed non-fattening and with no harmful additives. As there's no adventure which causes more questions in my mailbag than Wintersoft's *Juxtaposition,* I've decided it's about time to give a solution to the game, as it's been out long enough now for this not to spoil the adventure too much — being a solution, I'll have to print everything forwards rather than backwards, so I suggest you read only as far as the point you've reached in the adventure and see how to get past the problem there that's stumping you. If you read further and see something you didn't ought to, then you've only got yourself to blame.

Several readers have sent in solutions and part-solutions to this game, and thanks in particular to Richard Read of Manchester for his notes, but the one that was most detailed, being a step-by-step guide, came from adventurer extraordinaire, Simon Hargrave of Gloucestershire. Thanks to Simon then for the following, though note that you'll still need to map out the various areas.

In Baron White's City

Look at the bed and get everything, then tie the blue and red sheets together to make a rope, press button on vidphone to unlock door, wait till the droid is going away then open the door and go e-n-e-n-ne to the rail. Tie the rope to the rail and go down. Get the sleeping pills from the table and go to the shower room. LOOK TAP and TURN TAP then wait for the droid to arrive and short-circuit it. Get the blue key card from it. Leave via the door, go se-se-sw, unlock the door and go down to ground level. Go to the monorail platform (*don't* go into the endless rez zones!), wait for the car and board it when the doors open. Get the camera from the photo gallery. Only take the food from the dinner droid when you need to eat. Go to the ante-room and put the sleeping pills in the caretaker's coffee. Get the white key card and the cup from the caretaker and unlock the door. Get the breather mask, stickly tape and soldering iron. Go back to the street area and wear the mask to enter the airlock. When outside drop all but the mask, overalls, white ID tag and white key card, though you'll need to return for some items later. Go to Baron White's fort.

To Enter A Building

To do this you need to be exactly at the building, i.e. standing so that you cannot see it no matter which way you look, and then type ENTER.

Eating

At any time when you need to eat, enter a city and get the food from the food droid simply by asking it.

Nightfall

When night falls enter the nearest building and conserve strength this way. Night falls every twenty paces.

In Baron White's Fort

Give the white key card to Baron, take the red ID tag and go to Baron Red's fort.

In Baron Red's Lands

Get the brown ID tag from Baron Red's fort, then go to the shaft which contains the mine. Blind the night-eye droid by using the camera, take his spade and go through the door. Dig for the red cetite ore. Take the ore to Baron Red and he will give you the green ID tag. Go to Baroness Green's fort.

In Baroness Green's Lands

Go to the fort and get the inner tube from the wheel. Repair it with the sticky tape. Go to a power pyramid and open the valve, then inflate the inner tube. Go to Baroness Brown's fort.

In Baroness Brown's Lands

Go to the fort, then go north-east to a shaft by the mountains. Wear the inner tube and swim the river. Get the deeds of Brown from the skeleton and use the sender to go to Baroness Brown's fort.

In Baroness Green's Lands

Go to the tower, get the emeralds and use the sender unit to escape. Give the emeralds to Baroness Green in exchange for the blue ID tag. Go back to the river in Baroness Brown's lands and cross the river into Baron Blue's lands.

In Baron Blue Lands

Go to Blue's fort and get the film. Go to the green tower in Blue's lands and repair the panel with the soldering iron and the flux. Load the camera with the film and photograph the VDU message. Take this to Baron Blue for the yellow ID tag.

In Baron Yellow's Lands

Go to the fort and get the vacuum box. Go to the power pyramid, which has a shaft going down. Wear the overalls, mask and cyrobelt, carry the vacuum box and descend the shaft. Get the phoenix egg from beside the lava stream and put it in the vacuum box. Go back to Baron yellow and give him the egg in exchange for the pink ID tag.

In Baroness Pink's Lands

Go to the Hydroponics Dome, wear the magnetic boots and go up the ventilator shaft and get the oil in the cup. Get the seed from behind the panel. Oil the rusty droid and ask him to go west then unlock the door. Go to Baroness Pink's fort and get the weedkiller then go through the door which the droid unlocked. When you are entwined by the weeds then spray them with the weedkiller. Plant the seed in the growbag. Take the orchid to Baroness Pink in exchange for the black ID tag.

In Baroness Black's Lands

Go to the fort, open the chest and diffuse the bomb by pressing the buttons in order: blue-red-green-red-blue. Take the sheet of music and give it to the music droid, whereupon he will play the musical notes which unlock the door. Enter the room containing the 'Juxtaposition'. Do what the sign on the wall says and type SPELL.

Congratulations!

At least this is what you should now be told, along with the information that the adventure will be continued in Part II, *The Usurper of Rune.* You remember Part Two, it was the one that was going to be released by Easter...Easter 1986, that is. Never mind, I'm sure it will be with us eventually, and very welcome to, judging by the numbers of people who seem to have bought *Juxtaposition*.

The above solution, incidentally, isn't the only way of solving the game, as obviously there are a few things that can be done in different orders, so don't write in and say that your solution is better! On the other hand, if you can explain where the brown fog is, which a reader asked about in the April issue of *Dragon User,* but which Simon 'The Solver' Hargrave couldn't find anywhere, then by all means write in and tell us.

Simon is contactable at Crawley Hill Farm, Uley, Dursley, Glos. GL11 5BH, and he also has a bit of information to add to last month's details about *Madness and the Minotaur,* where he's trying to get his score up from 210 to the maximum 240. He advises against using the save routine as this only saves your current location and not your character status, as well as apparently rendering the 'akhirom' ineffective against the hydra.

And th-th-th-th-that's all folks!

Adventure Contact

Adventure: Return of The Ring. **Problem:** How to bind the six units for the time ring? How to use the copper ring, tracker, money sack and white pass? **Name:** Philip Callaghan. **Address:** 54 Penkvale Road, Moss Pit, Stafford, Staffs ST17 9EY.

Adventure: Williamsburg 3. **Problem:** Has anyone got a map? I need one. **Name:** Colin Mills. **Address:** 31 Leabank Drive, Northwick, Worcester WR3 7RA. **Adventure:** Vortex Factor. **Problem:** I need help. **Name:** Stuart Ellicott. **Address:** 65 Yewtree Cresc., Melton Mowbray, Leics LE13 1LN.

Adventure: 1) Shenanigans 2) Return Of The Ring. **Problem:** 1) How to get out of the residential area 2) How to get more gesal level. **Name:** D. Hamilton. **Address:** 10 Norris Way, Formby, Merseyside L37 8DB.

Adventure: Operation Saeras. **Problem:** After meeting a man called Rhythma it asks you to type in the code. I have no idea what the code is, please help me. **Name:** Suzanne Corne. **Address:** 9 Beehive Road, Goffs Oak, Herts EN7 5NL.

Adventure: Madness and the Minotaur. **Problem:** Can't find the mushroom, and how to get the light downstairs. **Name:** Endre Knudsen. **Address:** Hagneset, 6390 Vestnes, Norway.

Adventure: Juxtaposition. **Problem:** How do you get the Eridana Emerald? Where are the Deeds of Brown? **Name:** Steven Allan. **Address:** 45 Waltham Road, Overton, Hants RG25 3NQ.

Adventure: Castle of the Skull Lord. **Problem:** How to progress from the first six locations when there are no exits? **Name:** John Campbell Rees. **Address:** 12 Stuart Street, Treherbert, Rhondda, Mid Glamorgan CF42 5PR.

Adventure: 1) The Cricklewood Incident 2) Lost In Space. **Problem:** 1) How do you get past the white rabbit 2) Everything — can't get started. **Name:** Ian Thomas. **Address:** 45 Kirkfield Road, Darlington, Co. Durham, DL3 OAD.

Adventure: 1) Pyramid 2) Madness & The Minotaur. **Problem:** 1) How do you get past the Dragon? 2) How do you get past the false wall? **Name:** Daniel Lawrance. **Address:** 26 Lea Street, Lindley, Huddersfield, W. Yorks HD3 3LS.

Adventure: Wings of War. **Problem:** How do I get past the man with the lunger gun and how many things are there to collect in the forest and what are they? **Name:** Andrew Cooper. **Address:** 16 Oldbury St., Wednesbury, West Midlands WS10 0QJ.

Adventure Contact

To help puzzled adventurers further, we are instituting an Adventure Helpline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpline, 12/13 Little Newport Street, London WC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Trial to write to as well!

Adventure ....

Problem ....

....

Name ....

Address ....

....

Adding Anagrams

***Gordon Lee** takes a number of letters and sums them up.*

IN THE letters section of the July issue of 'Dragon User', reader Denis O'Mulloy, who describes himself as an avid fan of the competition page (Thanks, Denis!), asks for further details regarding the correct method to be used in solving the competition problems.

From time to time on the competition page, general remarks on problem solving — either generally or related to specific problems — have been given. Readers with back numbers will be able to check these out; readers without — tough luck!

Some hints on problem solving were given in the January '84 issue, and points relating to some specific competition questions were outlined in the February '84 and October '84 issues. The June/July competition pages contained information on computations involving large numbers of digits, the November '85 page had some digital shortcuts, while problem solving by the use of algorithms was covered in the March '86 Dragon User.

The Dragon, in common with other computers has its own specific peculiarities which can result in errors creeping in if precautions are not taken during programming. In May '84, exponential notation was mentioned, while problems of mathematical accuracy, and an unexpected feature of the Dragon's STR& command were noted in the September '84 issue. Some general comments on the pitfalls associated with the use of the trigonometric and random number functions appeared in the January '85 and June '86 issues respectively.

Mr O'Mulloy's term 'correct method' *is* a tougher nut to crack, since there may well be a number of different ways of reaching the final answer, and, so long as that answer is the correct one, which is the correct method? In one recent competition, the program listings included by competitors ranged from one on less than half a page at one extreme, to a six-foot long print-out at the other! No doubt both were 'correct' in that they produced the required answer, but the one was certainly more concise than the other. Perhaps, as a rule of thumb, a program which provides the correct answer, and in which the total programming time and running time is the shortest, should be regarded as the best.

Achieving this is really a matter of knowing your computer, and being aware of any short cuts which can speed up running times without affecting accuracy. This is best gained by experience, and in this respect I would recommend attempting the weekly competition which appears in Dragon User's sister paper, *Popular Computing Weekly*. The solutions, and a working listing are usually given, so even if readers do not actually submit solutions, they can always use these problems as tests of technique.

For readers who wish to delve more deeply into this field, I would recommend *Adventures with your Computer* by Lennart Rade and David Nelson (Penguin Books).

And now to this month's competition:

Professor Otto Hex was dining with two of his colleagues, Professor Bumble, the famous musicologist, and the mathematician Dr. Browze.

'I remember you saying,' mussed Hex to Bumble, 'that you once remarked that an anagram of Elgar was 'regal', an apt description of much of that composer's work. Well, I've found some more anagrams and I have constructed them into an alphamatic puzzle,'

He passed a slip of paper to Dr. Browze on which was written:

```
ELGAR
REGAL
LARGE
LAGER
GLARE
******
```

'It's a simple addition sum, and what you have to do is to find suitable values for the letters E, L, G, A, and R, such that the result is a six-digit sum, as represented by the row of stars. However, each of the digits in this sum must be found in the substitution above, so that they can all be re-substituted for letters to obtain a six-letter sequence. it need not represent an actual word, so you could have, for example, sequences such as LAAGEL or even GERLLG.

There are many answers posible, but can you find the one that I am thinking of if I tell you that the two examples given above are as far wrong as you can get, in that neither of these has a single letter in its correct place (so the first letter cannot be an 'L' or a 'G', and so on).

Professor Bumble looked for help to Dr. Browze. Can you help?

Prize

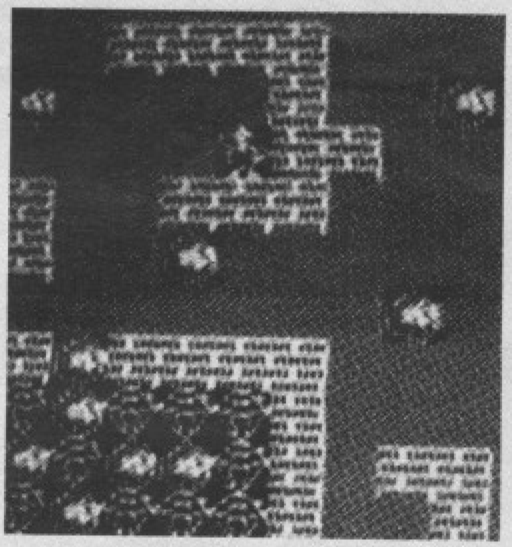
IN keeping with the high intellectual tone set by our competitions, this month's prizewinners will each receive a copy of *Boulder Crash*, by Blaby Computer Games, of whom our reviewer said "It's really unfair of you to send me a game ... that I spend so much time playing it I don't get round to reviewing it." and, further, "the slime ... Oh god, the slime!"

So get calculating for one of those twenty boulder clashes. One clue: the answer is not LLLLG.

Rules

Send us your answer, accompanied by a printout, or a handwritten copy, of the program you used to calculate it. No cassettes, please. Please mark the envelope OCTOBER COMPETITION to ensure it arrives at the right drawer, and remember amid all the excitement to include your name and address — if you must put it in the printout, how about upper case?

Because you're all so brilliant, we need a crashingly thrilling tie breaker to sift out the final twenty. This month's phrase is "Look out for falling rocks, but don't worry ...". Complete that in any way you like - we might like it too.



Boulder Crash

July Winners

All ye who are entered here will receive a copy of chart title *Moon Cresta*, by Incentive Software: Simon Aubrey of Swindon, A. Bussell of Clevedon, Michael Graham of Bangor, D. J. James of Swansea, J. Leyland of Grantham, M. J. Stentiford of Kingskerswell, Mike Hides of Sheffield, Nils Lindgren of Sweden, Robert Margrave of Leamington Spa, Graham Smith of Bristol, Christopher James of Malta, E. C. Hasted of Erith, Jonathan Harrop of Wantage, G. R. Barber of Sutton Coldfield, M. Owens of Pontypridd, Justin Hewitt of Hixon, F. J. Willers of Yarnfield, D. J. Gray of Cleveland, John A. Pollock of Stoke-on-Trent, and Phil Sapiro of Liverpool. Favourite saying this month: "The Dragon takes a giant step — the giant falls through a gap in the stairs." Solution

Solution

The solution to the problem if you follow all the instructions carefully is: the first grey slab you must step onto is "West edge, 58th slab from the north west corner", or, of course, "West edge, 44th slab from the south west corner" - depending which way you counted.